Journal of Machine Learning Research 9 (2008) 1295-1342 Submitted 7/07; Revised 2/08; Published 7/08

Using Markov Blankets for Causal Structure Learning

Jean-Philippe Pellet[*] JEP@ZURICH.IBM.COM
André Elisseeff AEL@ZURICH.IBM.COM
Data Analytics Group
IBM Zurich Research Laboratory
Säumerstraße 4, CH–8803 Rüschlikon

Editor: David Maxwell Chickering

Abstract

We show how a generic feature-selection algorithm returning strongly relevant variables can be turned into a causal structure-learning algorithm. We prove this under the Faithfulness assumption for the data distribution. In a causal graph, the strongly relevant variables for a node X are its parents, children, and children's parents (or spouses), also known as the Markov blanket of X. Identifying the spouses leads to the detection of the V-structure patterns and thus to causal orientations. Repeating the task for all variables yields a valid partially oriented causal graph. We first show an efficient way to identify the spouse links. We then perform several experiments in the continuous domain using the Recursive Feature Elimination feature-selection algorithm with Support Vector Regression and empirically verify the intuition of this direct (but computationally expensive) approach. Within the same framework, we then devise a fast and consistent algorithm, Total Conditioning (TC), and a variant, TC_{bw}, with an explicit backward feature-selection heuristics, for Gaussian data. After running a series of comparative experiments on five artificial networks, we argue that Markov blanket algorithms such as TC/TC_{bw} or Grow-Shrink scale better than the reference PC algorithm and provides higher structural accuracy.

Keywords: causal structure learning, feature selection, Markov blanket, partial correlation, statistical test of conditional independence

1. Introduction

In this paper, we are interested in using concepts from the feature-selection field to help causal structure learning. Causal structure learning (Pearl, 2000; Spirtes et al., 2001) is a multivariate data-analysis approach that aims to build a directed acyclic graph (DAG) showing direct causal relations among the variables of interest of a given system. These so-called causal graphs can be used together with dedicated rules called *do*-calculus (Pearl, 1995) to predict the effect of interventions, that is, of structural changes in the data-generating process. In this sense, it differs significantly from traditional machine-learning techniques: given a set of interventions, we can predict the behavior of a set of variables whose joint probability distribution has changed since the model was trained.

Building the causal graph is a difficult task, subject to a series of assumptions, and provably correct algorithms have an exponential worst-case complexity. Identifying the exact causal graph is in general impossible. By means of non-interventional data, causal graphs can only be identified up to *observational equivalence*: only adjacencies and so-called V-structures (two independent causes

∗. Also at Pattern Analysis and Machine Learning Group, Swiss Federal Institute of Technology Zurich, Universitätstraße 6, CH–8092 Zurich.

leading to the same effect) can be specified exactly (Pearl, 2000, p. 19). Typical structure-learning algorithms thus return partially directed acyclic graphs (PDAGs). These algorithms can be roughly classified into two categories: the *score-based* algorithms associate a score function with a DAG or PDAG given a training data set and perform, for instance, a greedy search in the space of DAGs or PDAGs (e.g., the GES algorithm, Chickering, 2002); the *constraint-based* algorithms look for dependencies and conditional dependencies in the data and build the causal graph accordingly. Well-known examples are the PC (Spirtes et al., 2001) or the IC (Pearl and Verma, 1991) algorithms. In an effort to get the best of both worlds, other algorithms use both conditional-independence tests and scores to build the network; MMHC (Tsamardinos et al., 2006) is such an example.

The range of data sets that the typical algorithms can deal with is restricted: not any probability distribution can be *faithfully* represented by a DAG. Faithfulness of the distribution is a well-defined condition: it guarantees the existence of a DAG, called a *perfect map*, where there is a one-to-one mapping between the graphical criterion of d-separation and conditional independence in the data.[1] Nilsson et al. (2007) discuss faithful distributions and other types of distributions with respect to properties of conditional independence. In the literature, Faithfulness is a precondition to prove correctness of the algorithms.

In practice, both existing score-based and constraint-based techniques deal primarily with discrete data sets. Score-based approaches for continuous variables are computationally expensive;[2] as for the constraint-based approaches, only the multivariate Gaussian case has been dealt with efficiently (Scheines et al., 1995). Margaritis (2005) proposed a distribution-free test of conditional independence, which is very computationally expensive and cannot be readily used with the current constraint-based algorithms for all but very small networks.

Coming from the machine-learning community, feature selection (John et al., 1994; Guyon and Elisseeff, 2003) is a common technique that aims at reducing the number of variables or features used for building more efficient or more robust models. Techniques have evolved to be able to handle nonlinear relationships between variables, redundant variables, in both discrete and continuous domains. Feature selection and causal structure learning are related by a common concept: the *Markov blanket* of a variable X is the smallest set $\mathbf{Mb}(X)$ containing all variables carrying information about X that cannot be obtained from any other variable.[3] In feature selection, this is the set of *strongly relevant* features; that is, of features which carry information about the target that cannot be obtained from any other variable (Kohavi and John, 1997). In a causal graph, this is the set of all parents, children, and spouses of X. The feature-selection task and the causal graph construction task can both be stated to some extent as Markov blanket identification tasks.

Relating feature selection and causal structure learning is not new. Several algorithms identifying the Markov blanket of a single variable with techniques inspired from causal structure learning have been proposed as the optimal solution to the feature-selection problem in the case of a faithful distribution. Tsamardinos and Aliferis (2003) show that for faithful distributions, the Markov blanket of a variable Y is exactly the set of strongly relevant features, and prove its uniqueness. They propose the Incremental Association Markov Blanket (IAMB) algorithm to determine it. With the same Faithfulness assumption, the Min-Max Markov Blanket algorithm (MMMB) (Tsamardinos

1. Conditional independence and d-separation are defined formally in Section 2.
2. Computationally tractable methods to learn Bayesian networks from continuous data exist (Fu, 2005), like Bach and Jordan (2003), but do not offer the causality-related theoretical correctness guarantees.
3. Some authors write "Markov blanket" without the notion of minimality, and use "Markov boundary" to note the smallest Markov blanket $\mathbf{Mb}(X)$. Even if defined as minimal, $\mathbf{Mb}(X)$ is generally not unique.

et al., 2003) identifies the Markov blanket of a variable Y by calling a subroutine Min-Max Parents and Children (MMPC). This subroutine finds the direct parents and children of Y with association measures and conditional-independence tests. MMPC is again called on each of these nodes to find potential spouses of Y. False positives are then discarded with conditional-independence tests. MMMB was further discussed by Peña et al. (2005), who propose AlgorithmMB, a similar approach based on scores and conditional-independence tests to retrieve $\mathbf{Mb}(Y)$. The HITON_MB algorithm (Aliferis et al., 2003) is similar in its main steps, and selects an optimal subset of the Markov blanket of a target variable given the Faithfulness assumption. Nilsson et al. (2007) also propose a theoretical algorithm for consistent identification of strongly relevant features in polynomial time for the class of strictly positive distributions. They also argue that some common backward feature-elimination algorithms like Recursive Feature Elimination (Guyon et al., 2002) are actually consistent, in the sense that they return the set of strongly relevant features in the large sample limit.[4]

These are examples of using causal structure learning or similar constraint-based techniques to help feature selection (see Guyon et al., 2007, for a review of those techniques). In this paper, we propose a framework to do the converse. We present a generic approach using the outcome of a consistent feature-selection algorithm *FS* to build an approximate structure of the true causal graph. If we assume that *FS* returns the Markov blanket of the variables, we can show how to turn this approximate result, called *moral graph* (Lauritzen and Spiegelhalter, 1988), into a provably correct PDAG depicting the causal structure. This approach is also used in the Grow-Shrink algorithm (Margaritis and Thrun, 1999), which also builds a moral graph before adjusting the local structure.

This paper contributes a generic algorithm to build a causal graph which clearly separates the Markov blanket identification and the needed local adjustments, an efficient algorithm to perform those adjustments, and two fast instances of the generic algorithm for multivariate Gaussian data sets. This is presented as follows: in Section 2, we first review the needed terms and definitions from feature selection and causality. In Section 3, we make the link from the outcome of a feature-selection algorithm to a causal graph by detailing the needed local adjustments and detail an efficient way to perform them. We directly apply it in Section 4, where we describe how we can build causal graphs using the RFE feature-selection algorithm. As this direct application is very computationally intensive, we then show our more efficient instantiations of the generic algorithm optimized for the multivariate Gaussian case, the TC and TC_{bw} algorithms. We list our experimental results in Section 5, showing through empirical evaluation that Markov blanket algorithms are more scalable and more accurate than the reference PC algorithm. We finally conclude in Section 6 and list proofs in Appendix A.

1.1 Notation

Boldface capitals designate either matrices or sets of random variables or nodes in a graph, depending on the context. $\mathbf{V}$ is the set of all variables in the analysis. Italicized capitals like X, Y, Z are random variables or nodes and elements of $\mathbf{V}$. Vectors are set in boldface lowercase, as $\mathbf{b}$ or $\mathbf{w}$; scalars in italics, as the number of samples n or the number of variables (the problem dimension) d. We indiscriminately write "variable" or "feature" to refer to any variable in the causal analysis or

4. Actually, their definition of consistency has to do with returning the set of features relevant to the Bayes classifier, which is slightly stronger than strong relevance as used here.

any node in a causal graph, and write "predictor" to designate a variable used as input for a given classifier or regression model.

2. Background

We formalize the feature-selection task suited for our needs and provide relevant definitions. We do the same for the causal structure-learning task and prepare the needed basis for drawing the parallels between the two in the next section.

2.1 Feature Selection

We are given a data set of n samples $D = \{(\mathbf{x}_i, y_i),\ 1 \leq i \leq n\}$. Each data point $(\mathbf{x}_i, y_i)$ has $d-1$ inputs, modeled as a vector $\mathbf{x}_i \in \mathbb{R}^{d-1}$, and an output, or *target*, $y_i \in \mathbb{R}$ (we use $d-1$ and not d for the size of $\mathbf{x}_i$ for consistency with the rest of the paper). The data points are assumed to be drawn i.i.d. from a joint probability distribution over the random variables $\mathbf{V} = \mathbf{X} \cup \{Y\}$. The result of the feature-selection task we are interested in is a set of retained variables $\mathbf{F} \subseteq \mathbf{X}$. How many variables to retain and which variables to retain depends on the particular algorithm, and usually maximizes some tradeoff between efficiency and classification/regression error of a given learning task.

John et al. (1994) propose a classification of the input variables $\mathbf{X}$ with respect to their relevance to the target Y in terms of *conditional independence*.

Definition 1 (Conditional independence) *In a variable set* $\mathbf{V}$*, two random variables* X, Y *are* conditionally independent *given* $\mathbf{Z} \subseteq \mathbf{V} \setminus \{X, Y\}$*, noted* $(X \perp\!\!\!\perp Y \mid \mathbf{Z})$*, if:*

$$\forall x, y, \mathbf{z} : P(X = x \mid Y = y, \mathbf{Z} = \mathbf{z}) = P(X = x \mid \mathbf{Z} = \mathbf{z}),$$

provided that $\forall \mathbf{z} : P(\mathbf{Z} = \mathbf{z}) > 0$.

Conditional independence is a generalization of the traditional notion of statistical independence. If two variables X and Y are independent, then the joint distribution is the product of the marginals: $P(X = x, Y = y) = P(X = x)P(Y = y)$. If they are dependent given some conditioning set $\mathbf{Z}$, then we can write $P(X = x, Y = y \mid \mathbf{Z} = \mathbf{z}) = P(X = x \mid \mathbf{Z} = \mathbf{z})P(Y = y \mid \mathbf{Z} = \mathbf{z})$. Conditional independence is a key concept in Bayesian networks (Pearl, 1988) because of the factorizations of the joint probability distribution it allows.

In feature selection, relevance of predictors to the target as proposed by John et al. (1994) is expressed in terms of conditional independence. In the following definitions, we write X_i to note the ith input variable, and $\mathbf{X}_{\setminus i}$ to note all input variables but the ith one.

Definition 2 (Strong relevance) *A variable* X_i *is* strongly relevant *to the target* Y *if*

$$P(Y \mid \mathbf{X}_{\setminus i}) \neq P(Y \mid \mathbf{X}_{\setminus i}, X_i).$$

A variable is strongly relevant to the target if it carries information about Y that no other variable carries. This is expressed in the definition by a change in the probability distribution of the target between conditioning on all other variables, $\mathbf{X}_{\setminus i}$, and also including X_i in the conditioning set. If X_i carries no exclusive information about Y, the two distributions will be identical.

Definition 3 (Weak relevance) *A variable X_i is* weakly relevant *to the target Y if it is not strongly relevant and*

$$\exists \mathbf{S} \subseteq \mathbf{X}_{\setminus i} : P(Y \mid \mathbf{S}) \neq P(Y \mid \mathbf{S}, X_i).$$

We speak of weak relevance of a variable X_i when there exists a certain context $\mathbf{S}$ in which it carries information about the target. However, this is not necessarily exclusive information, as it may be possible to obtain it from other variables.

Corollary 4 (Irrelevance) *A variable X_i is* irrelevant *to the target Y if it is neither strongly nor weakly relevant, that is, if*

$$\forall \mathbf{S} \subseteq \mathbf{X}_{\setminus i} : P(Y \mid \mathbf{S}) = P(Y \mid \mathbf{S}, X_i).$$

A variable is irrelevant if carries no information about the target at all, no matter what the context is.

For our purposes, we assume that the feature-selection algorithm returns the set of all strongly relevant variables, and only those.[5] (In Section 5, we discuss with experiments whether this is a reasonable assumption with the RFE algorithm.) Put in terms of conditional independence, the result $\mathbf{F}_Y$ of the feature-selection task with target Y is, with $\mathbf{V} = \mathbf{X} \cup \{Y\}$:

$$\mathbf{F}_Y = \left\{ X \;\middle|\; (X \not\perp\!\!\!\perp Y \mid \mathbf{V} \setminus \{X, Y\}) \right\}. \tag{1}$$

That is the set of the variables that are dependent on the target Y, conditioned on all others. We need this property in Section 3 to use the output of the feature-selection task to build a causal graph. Note that if we repeat the feature-selection task using as target another variable $X \in \mathbf{V}$ yielding a result $\mathbf{F}_X$, we have:

$$X \in \mathbf{F}_Y \iff Y \in \mathbf{F}_X. \tag{2}$$

This follows as a direct consequence of (1) due to the symmetry of the conditional-independence relation $(X \perp\!\!\!\perp Y \mid \mathbf{Z})$ with respect to X and Y.

2.2 Causal Structure Learning

In causal structure learning, we are interested in representing graphically conditional dependencies found in the data. Under a set of assumptions, they have a causal interpretation. For this task, we have a data set of n samples $D = \{\mathbf{v}_i, 1 \leq i \leq n\}$. We do not designate a specific target variable in $\mathbf{V}$ as we are interested in learning the full structure of the network.

The graphical representation of choice for causal models is directed acyclic graphs (DAGs) (Pearl, 2000). In a causal graph represented by a DAG, we want to represent direct causal relations with arcs between pairs of variables. Choosing DAGs for this task implies restrictions, an obvious one of which is that causal feedback loops are excluded from the analysis. More formally, the joint probability distribution has to be *faithful* (or *DAG-isomorphic*, Pearl, 1988, p. 128); that is, there must exist a DAG that represents all (conditional) dependencies and independencies entailed by the distribution. Such a graph is called a *perfect map* of the distribution if there is a one-to-one mapping between the conditional-independence relation defined on variables and the *d-separation criterion* defined on the graphical nodes.

5. In the general case, this set can be empty without excluding the existence of other weakly relevant variables (Tsamardinos and Aliferis, 2003). In the next subsection, we detail the Faithfulness hypothesis, which allows us to exclude this particular case.

Definition 5 (*d*-separation) *In a DAG $\mathcal{G}$, two nodes X, Y are* d-separated *by $\mathbf{Z} \subseteq \mathbf{V} \setminus \{X, Y\}$, written $(X \dashv\vdash Y \mid \mathbf{Z})$, if every path from X to Y is blocked by $\mathbf{Z}$. A path is* blocked *if at least one diverging or serially connected node is in $\mathbf{Z}$ or if at least one converging node and all its descendants are not in $\mathbf{Z}$. If X and Y are not* d-*separated by $\mathbf{Z}$, they are* d-*connected: $(X \leftrightarrow Y \mid \mathbf{Z})$.*

Determining whether two nodes in a graph are *d*-separated given some conditioning set is not visually immediate. It may for instance be unintuitive that whereas conditioning on a node W on a directed path $X \to W \to Y$ blocks the path from X to Y, conditioning on a common child Z of two variables X, Y in $X \to Z \leftarrow Y$ connects them. In the latter case, this common child is called a *collider*. If, furthermore, two parents X, Y of a node Z are nonadjacent in the full graph, then Z is called an *unshielded collider* for the pair (X, Y).

The definition of *d*-separation was worked out by Pearl (1988) to match as closely as possible the complicated nature of the conditional-independence relation with a graphical criterion, so that the class of faithful distributions, which can be represented by a perfect map, is as large as possible, while still keeping a natural graphical representation.

Definition 6 (Perfect map) *A DAG $\mathcal{G}$ is a* directed perfect map *of a joint probability distribution $P(\mathbf{V})$ if there is bijection between* d-*separation in $\mathcal{G}$ and conditional independence in P:*

$$\forall X, Y \in \mathbf{V}, \forall \mathbf{Z} \subseteq \mathbf{V} \setminus \{X, Y\} : \big((X \dashv\vdash Y \mid \mathbf{Z}) \iff (X \perp\!\!\!\perp Y \mid \mathbf{Z}) \big). \tag{3}$$

If we take apart the perfect-map equivalence, we distinguish two important concepts, known as the Causal Markov condition and the Faithfulness condition (Spirtes et al., 2001, p. 29).

The **Causal Markov condition** is said to hold for a graph $\mathcal{G} = \langle \mathbf{V}, \mathbf{E} \rangle$ and a probability distribution $P(\mathbf{V})$ if every variable is statistically independent of its graphical non-descendants (intuitively, of its non-effects, direct or indirect) conditional on its graphical parents (intuitively, its direct causes) in P. Pairs $\langle \mathcal{G}, P \rangle$ that satisfy the Causal Markov condition satisfy the implication

$$\forall X, Y \in \mathbf{V}, \forall \mathbf{Z} \subseteq \mathbf{V} \setminus \{X, Y\} : \big((X \dashv\vdash Y \mid \mathbf{Z}) \implies (X \perp\!\!\!\perp Y \mid \mathbf{Z}) \big).$$

This is called *I-map property* by Pearl (1988).

The **Faithfulness condition** can be interpreted as the converse of the Causal Markov condition, and states that the only conditional independencies to hold are those specified by the Causal Markov condition:

$$\forall X, Y \in \mathbf{V}, \forall \mathbf{Z} \subseteq \mathbf{V} \setminus \{X, Y\} : \big((X \leftrightarrow Y \mid \mathbf{Z}) \implies (X \not\perp\!\!\!\perp Y \mid \mathbf{Z}) \big).$$

If the Causal Markov and Faithfulness conditions hold together for a pair $\langle \mathcal{G}, P \rangle$, then we find again the equivalence (3), and $\mathcal{G}$ is a perfect map of P.

In practice, the Causal Markov condition is used by the so-called constraint-based algorithms to perform conditional-independence tests on the data and build the graph accordingly, and Faithfulness is assumed to prove that the graph is correct. Hausman and Woodward (1999) discuss and explain in more detail the Causal Markov condition, and Steel (2005) discusses the Faithfulness condition and its motivations, pointing out cases where it can be violated. While the former is in general not violated simply by construction of the causal graph, violation of the latter occurs if the probability distribution is not faithful. A simple example is the n-bit parity problem where the prior probability of each bit is uniform, of which the XOR problem is a special case: each variable is

unconditionally independent of every other, but any variable pair becomes dependent conditioned on all other variables. On this problem, current constraint-based algorithms yield an empty graph because of the pairwise unconditional independencies, although it is not true that the data shows no dependency at all since one variable is a well-defined function of all others.

From this point on and for all proofs, we assume that the working data set D has a distribution that does not violate Faithfulness, and that it can thus be represented by a perfect map. In such a context, however, it is still not clear that causation can be inferred from conditional independence. We now proceed to explain the relation between causation and conditional independence.

Assuming Faithfulness, direct causation between X and Y, noted $X \dashrightarrow Y$, implies that X and Y are dependent given any conditioning set (Pearl and Verma, 1991, see definitions of potential and genuine causes):

$$X \dashrightarrow Y \implies \big(\forall \mathbf{S} \subseteq \mathbf{V} \setminus \{X,Y\} : (X \not\perp\!\!\!\perp Y \mid \mathbf{S})\big).$$

We denote the absence of direct causation by $X \not\dashrightarrow Y$. The exact converse of this implication does not hold. If we make the **Causal Sufficiency assumption** (Spirtes et al., 2001), that is, assume that no hidden common cause of two variables exists, we can write:

$$\big(\forall \mathbf{S} \subseteq \mathbf{V} \setminus \{X,Y\} : (X \not\perp\!\!\!\perp Y \mid \mathbf{S})\big) \implies X \dashrightarrow Y \text{ or } Y \dashrightarrow X. \tag{4}$$

Using (4), we can theoretically determine all adjacencies of the causal graph with conditional-independence tests, but we cannot orient the edges. But there is a special causation pattern where conditional-independence tests can reveal the direction of causation. It is known as a **V-structure** (Pearl, 2000): two common causes X, Y, initially independent,[6] become dependent when conditioned on a common effect Z, then acting as a collider. This is noted $X \dashrightarrow Z \dashleftarrow Y$, where we also require $X \not\dashrightarrow Y$ and, symmetrically, $Y \not\dashrightarrow X$. Formally, we have:

$$\begin{aligned} X \dashrightarrow Z \dashleftarrow Y \text{ and } X \not\dashrightarrow Y \text{ and } Y \not\dashrightarrow X \\ \implies \big(\exists \mathbf{S} \subseteq \mathbf{V} \setminus \{X,Y,Z\} : (X \perp\!\!\!\perp Y \mid \mathbf{S}) \text{ and } (X \not\perp\!\!\!\perp Y \mid \mathbf{S} \cup \{Z\})\big). \end{aligned}$$

The exact converse does not hold either. But using (4), we can find an equivalence relation defining a V-structure, still assuming Causal Sufficiency: first, we certify the existence of a link between X and Z and between Y and Z. Z is then identified as an unshielded collider if conditioning on it creates a dependency between X and Y:

$$\begin{aligned} X \dashrightarrow Z \dashleftarrow Y \iff \Big(&\big(\exists \mathbf{S} \subseteq \mathbf{V} \setminus \{X,Y,Z\} : (X \perp\!\!\!\perp Y \mid \mathbf{S}) \text{ and } (X \not\perp\!\!\!\perp Y \mid \mathbf{S} \cup \{Z\})\big) \\ &\text{and } \big(\forall \mathbf{S} \subseteq \mathbf{V} \setminus \{X,Z\} : (X \not\perp\!\!\!\perp Z \mid \mathbf{S})\big) \\ &\text{and } \big(\forall \mathbf{S} \subseteq \mathbf{V} \setminus \{Y,Z\} : (Y \not\perp\!\!\!\perp Z \mid \mathbf{S})\big)\Big). \end{aligned} \tag{5}$$

Actually, typical algorithms first establish the existence of a link between two variables by seeking a certificate equivalent to, or implicating the premise of, (4), and then look for orientation possibilities. Note that there is no guarantee that all links can be oriented into causal arcs, and that in

6. The two causes X and Y actually do not need to be unconditionally independent, but there must exist a (possibly empty) separating set $\mathbf{S}_{XY} \subseteq \mathbf{V} \setminus \{X,Y\}$ such that $(X \perp\!\!\!\perp Y \mid \mathbf{S}_{XY})$ for the collider to be identifiable. This implies that no direct causation $X \dashrightarrow Y$ or $Y \dashrightarrow X$ may exist: the collider must be unshielded.

general we therefore cannot recover the full causal structure with conditional-independence tests. This is the problem known as **causal underdetermination** (Spirtes et al., 2001, p. 62): for the structure-learning task given observational data, a correct graph is specified by its adjacencies and its V-structures only. Partially oriented graphs returned by structure-learning algorithms represent *observationally equivalent classes* of causal graphs (Pearl, 2000, p. 19). This means that for a given joint probability distribution $P(\mathbf{V})$, the set of all conditional-independence statements that hold in P does not yield a unique perfect map in general.

Formally, if we combine (3), (4) and (5), we find, for a perfect causal map $\mathcal{G}$ (using the symbol "$\multimap$" to denote direct causation and "$\rightarrow$" to denote an arc in the graph):

$$\begin{aligned} X,Y \text{ adjacent in } \mathcal{G} &\iff X \multimap Y \text{ or } Y \multimap X \\ X \rightarrow Z \leftarrow Y &\iff X \multimap Z \leftarrow Y. \end{aligned} \tag{6}$$

It is sometimes possible to orient further arcs in a graph by looking at already-oriented arcs and propagating constraints, preventing acyclicity and the creation of additional V-structures other than those already detected. The graph after this constraint-propagation step is called *completed PDAG*, *maximally oriented PDAG* (CPDAG), or *essential graph*, depending on the author.

3. Causal Network Construction Based on Feature Selection

We have looked at the ideal outcome of feature selection in (1) and how to read a causal graph in (6). We now turn to showing how feature selection can be used to build a causal graph. From now on and for the rest of this paper, we assume that the joint probability distribution over all variables $\mathbf{V}$ is faithful.

3.1 Identifying the Markov Blankets

In the context of directed graphical models, the Markov blanket of a node X, noted $\mathbf{Mb}(X)$, is the set of parents, children, and children's parents (spouses) of X. As an easy property, note that we have:

$$X \in \mathbf{Mb}(Y) \iff Y \in \mathbf{Mb}(X).$$

The following statement is a key property of Markov blankets.

Property 7 (Total conditioning) *In the context of a faithful causal graph $\mathcal{G}$, we have:*

$$\forall X,Y \in \mathbf{V}: \left(X \in \mathbf{Mb}(Y) \iff (X \not\perp\!\!\!\perp Y \mid \mathbf{V} \setminus \{X,Y\})\right).$$

(See Appendix A for the proof.) This property says that the Markov blanket of each node is the set of all variables that are dependent on it, conditioned on all other variables. In other words, in a causal graph, the parents, children, and spouses of Y store information about Y that cannot be obtained from any other variable. Note that $\mathbf{Mb}(Y)$ then has exactly the property of the output of feature selection, $\mathbf{F}_Y$, as characterized in (1). This links feature selection and causal structure learning in the sense that

$$\mathbf{F}_Y = \mathbf{Mb}(Y),$$

the Faithfulness assumption guaranteeing the unicity of $\mathbf{Mb}(Y)$. However, Markov blankets alone do not fully specify a causal graph. Thus, feature selection, even if guaranteed to find only strongly relevant features, cannot be directly used to construct the graph as we want it to be. The problem is that spouses of Y, even if not directly linked in the original graph, would be linked in $\mathbf{F}_Y$ and $\mathbf{Mb}(Y)$. An additional step is needed to transform the Markov blankets into parents, children, and spouses.

3.2 Recovering the Local Structure

The result of feature selection can be graphically shown by an undirected graph $\mathcal{G} = \langle \mathbf{V}, \mathbf{E} \rangle$ where $(X,Y) \in \mathbf{E} \Leftrightarrow X \in \mathbf{F}_Y$. This graph is close to the original causal graph in that it contains all arcs as undirected links, and additionally links spouses together, and is called the *moral graph* of the original directed graph (Lauritzen and Spiegelhalter, 1988, p. 166). The extra step needed to transform this graph into a causal PDAG is the deletion of the spouse links and the orientation of the arcs, a task which we call "resolving the Markov blankets."

An existing algorithm can resolve the Markov blankets, that is, use Markov blanket information to infer the local structure around a node: the Grow-Shrink (GS) algorithm, proposed by Margaritis and Thrun (1999). The full algorithm first finds the Markov blanket for each variable, and performs further conditional-independence tests around each variable to infer the structure locally. It then uses a heuristics to remove cycles possibly introduced by previous steps. We list in Algorithm 1 (using our notation) the steps of the algorithm responsible for building the local structure using the Markov blanket information, as this is exactly the task we are trying to solve. In the code, $\mathbf{Bd}(X)$ stands for the *boundary* of X; that is, the set of its direct neighbors in the graph $\mathcal{G}$. It is different from $\mathbf{Mb}(X)$ in that whereas $\mathbf{Mb}(X)$ is passed as input to the algorithm and is fixed, $\mathbf{Bd}(X)$ depends on the graph $\mathcal{G}$, which is modified throughout the algorithm. We note a conditional-independence test with a subroutine call to CONDINDEP$(X,Y,\mathbf{Z})$: ideally, this function returns *true* when $(X \perp\!\!\!\perp Y \mid \mathbf{Z})$ holds, and *false* otherwise. More will be said about the actual implementation of such tests in Section 4. The command **break** is used to break out of the innermost loop, saving unnecessary computations.

The GS algorithm makes two passes over all variables and the members of their Markov blankets (or direct neighbors in the second pass). It first removes the possible spouse links between linked variables X and Y by looking for a d-separating set around X and Y. In a second pass, it orients the arcs whenever it finds that conditioning on a middle node creates a dependency. While searching for the appropriate conditioning set, GS selects the smallest base search set (set $\mathbf{B}$ in Algorithm 1) for each phase. This has two very desirable effects. First, it reduces the number of tests, which is useful because each phase contains a subset search, exponential in time complexity with respect to the searched set. Second, it reduces the average size of the conditioning set, which increases the power of the statistical tests, and thus helps reduce the number of Type II errors.

While the GS approach considerably reduces the number of tests to be performed with respect to a large subset search, it is possible to perform fewer tests while still identifying correctly the structure and orienting the arcs, and decreasing the average conditioning set size. A helpful observation is that orientation and removal of the spouse links can be done together in a single pass. We know, as discussed in the previous section, that only arcs in V-structures can be oriented: fortunately, V-structures are exactly spotted when we identify a spouse link to be removed. Two spouses X and Y that are not directly linked in the original causal graph can be d-separated by some set of

Algorithm 1 Resolve the Markov Blankets with the Grow-Shrink Algorithm

```
1:  procedure ResolveMarkovBlankets_GrowShrink
    Input:  Mb(·) :  the Markov blanket information for each node X ∈ V
    Output:  G :  partially oriented DAG
    /* Compute graph structure */
2:    G ← moral graph according to Mb(·)
3:    for each X ∈ V and Y ∈ Mb(X) do
4:      B ← smallest set of {Bd(X) \ {Y}, Bd(Y) \ {X}}
5:      for each S ⊆ B do
6:        if CondIndep(X, Y, S) then remove link X − Y from G; break
7:      end for
8:    end for
    /* Orient edges */
9:    for each X ∈ V and Y ∈ Bd(X) do
10:     for each Z ∈ Bd(X) \ Bd(Y) \ {Y} do
11:       orient Y → X    /* to be corrected if a test yields independence */
12:       B ← smallest set of {Mb(Y) \ {Z}, Mb(Z) \ {Y}}
13:       for each S ⊆ B do
14:         if CondIndep(Y, Z, S ∪ {X}) then remove orientation Y → X; break
15:       end for
16:       if Y → X then break
17:     end for
18:   end for
19:   return G
20: end procedure
```

nodes. Thus, if we can find a set $\mathbf{S}_{XY}$ that makes X and Y conditionally independent, we know that the link between them is a spouse link to be removed. Moreover, we know that any node Z part of the intersection of their Markov blankets not included in $\mathbf{S}_{XY}$ is a collider and thus a common child, and that the triplet (X,Z,Y) is actually a V-structure $X \rightarrow Z \leftarrow Y$ in the original graph. This follows from the definition of d-separation. What we need is an efficient search algorithm to find such d-separating sets.

An approach based on this observation has two main benefits. First, it only searches the triangles, that is, the cliques of three nodes, in the moral graph. Assuming that the information about the Markov blanket is correct, only triangles can hide spouse links and V-structures. Second, for each connected pair $X - Y$ in a triangle, decisions about possible spouse links and arc orientation are taken together and thus faster.

Pseudocode for the proposed search algorithm is listed in Algorithm 2, where the notation $\mathcal{G}^{\setminus XY}$ denotes the moral graph $\mathcal{G}$ where all direct links involving X or Y have been removed. The algorithm uses the following concept.

Definition 8 (Collider sets) *In an undirected graph $\mathcal{G} = \langle \mathbf{V}, \mathbf{E} \rangle$, let* $\mathbf{Tri}(X-Y)$ *(with $X, Y \in \mathbf{V}$ and $(X,Y) \in \mathbf{E}$) be the set of vertices forming a triangle with X and Y:*

$$\mathbf{Tri}(X-Y) = \{Z \in \mathbf{V} \mid (X,Z) \in \mathbf{E}, (Y,Z) \in \mathbf{E}\}.$$

Suppose that $\mathcal{G}$ is the moral graph of the DAG representing the causal structure of a faithful data set. A set of vertices $\mathbf{Z} \subseteq \mathbf{Tri}(X-Y)$ then has the Collider Set *property for the pair (X,Y) if it is the largest set that fulfills*

$$\exists \mathbf{S}_{XY} \subseteq \mathbf{V} \setminus \{X,Y\} \setminus \mathbf{Z} : (X \perp\!\!\!\perp Y \mid \mathbf{S}_{XY}) \tag{7}$$

$$\text{and} \quad \forall Z_i \in \mathbf{Z} : (X \not\perp\!\!\!\perp Y \mid \mathbf{S}_{XY} \cup \{Z_i\}). \tag{8}$$

The set $\mathbf{S}_{XY}$ is then a d*-separating set for X, Y.*

Lemma 9 *In the context of a faithful causal graph, the set $\mathbf{Z}$ that has the Collider Set property for a given pair (X,Y) exists if and only if X is neither a direct cause nor a direct effect of Y. This set $\mathbf{Z}$ is unique when it exists. (Proof in Appendix A.)*

The purpose of Algorithm 2 is thus to find these collider sets (in the pseudocode, the symbol $\subsetneq$ denotes the strict subset relation). The algorithm loops over all triangle links and performs a collider set search for each of them. Let $X - Y$ be one of these links: if it is not a spouse link, the search procedure will leave the value of the d-separating set $\mathbf{S}_{XY}$ to its default value, **null**. Otherwise, $\mathbf{S}_{XY}$ will be set to a (possibly empty[7]) set for X and Y. The collider set can be inferred by removing the d-separating set from the triangle nodes $\mathbf{Tri}(X-Y)$: as $\mathbf{Tri}(X-Y)$ contains nodes on a path of length 2 between X and Y, finding a d-separating set that does not contain some of these nodes proves that they can only be colliders according to the definition of d-separation.[8] For instance, if the procedure produces an empty set for a given linked pair $X-Y$, then X and Y are unconditionally independent, and therefore all nodes in $\mathbf{Tri}(X-Y)$ are colliders.

Two caveats have to be observed during this search, however. First, there might be other active, d-connecting paths between X and Y that are not going through any node of $\mathbf{Tri}(X-Y)$. Those nodes must be blocked by appropriate conditioning on the boundary of X or Y as determined by the base conditioning set at line 6. Second, this base conditioning set must be checked not to include any descendant of possible colliders. If it did, it would open a d-connecting path according to Definition 5. This check is performed at lines 13 to 21. At line 13, we build a set $\mathbf{D}$ that includes all possible descendants of currently conjectured colliders that intersect our base conditioning set $\mathbf{B}$. The following loop makes sure none of them was opening a path between X and Y.

Theorem 10 *In the large sample limit, for faithful, causally sufficient data sets, the procedure* RESOLVEMARKOVBLANKETS_COLLIDERSETS *correctly identifies all V-structures and all spouse links, assuming consistent statistical tests. (Proof in Appendix A.)*

This procedure is best understood with a graphical example. Consider the sample local structure in Figure 1, imagine it is part of a larger network, and suppose we are performing the search

7. Note that returning an empty d-separating set in $\mathbf{S}_{XY}$ is different from returning **null**, signaling the absence of any such set.
8. The next paragraphs describe patterns where this is not true and show how the algorithm still deals with them correctly.

Algorithm 2 Resolve the Markov Blankets with Collider Sets

1: **procedure** RESOLVEMARKOVBLANKETS_COLLIDERSETS
Input: $\mathbf{Mb}(\cdot)$: the Markov blanket information for each node $X \in \mathbf{V}$
Output: $\mathcal{G}$: partially oriented DAG
2: $\mathcal{G} \leftarrow$ moral graph according to $\mathbf{Mb}(\cdot)$
3: $\mathbf{C} \leftarrow \{\}$, an empty list of orientation directives
4: **for each** edge $X - Y$ part of a fully connected triangle **do**
5: $\mathbf{S}_{XY} \leftarrow$ **null** /* search for d-separating set */
6: $\mathbf{B} \leftarrow$ smallest set of $\{\mathbf{Bd}(X) \setminus \mathbf{Tri}(X-Y) \setminus \{Y\}, \mathbf{Bd}(Y) \setminus \mathbf{Tri}(X-Y) \setminus \{X\}\}$
7: **for each** $\mathbf{S} \subsetneq \mathbf{Tri}(X-Y)$ **do** /* subset search */
8: $\mathbf{Z} \leftarrow \mathbf{B} \cup \mathbf{S}$
9: **if** CONDINDEP$(X, Y, \mathbf{Z})$ **then**
10: $\mathbf{S}_{XY} \leftarrow \mathbf{Z}$
11: **break** to line 23
12: **end if**
13: $\mathbf{D} \leftarrow \mathbf{B} \cap \left\{\text{nodes reachable by } W \text{ in } \mathcal{G}^{\setminus XY} \mid W \in \left(\mathbf{Tri}(X-Y) \setminus \mathbf{S}\right)\right\}$
14: $\mathbf{B}' \leftarrow \mathbf{B} \setminus \mathbf{D}$
15: **for each** $\mathbf{S}' \subsetneq \mathbf{D}$ **do** /* descendant of collider may be opening a path */
16: $\mathbf{Z} \leftarrow \mathbf{B}' \cup \mathbf{S}' \cup \mathbf{S}$
17: **if** CONDINDEP$(X, Y, \mathbf{Z})$ **then**
18: $\mathbf{S}_{XY} \leftarrow \mathbf{Z}$
19: **break** to line 23
20: **end if**
21: **end for**
22: **end for**
23: **if** $\mathbf{S}_{XY} \neq$ **null then** /* save orientation directive */
24: mark link $X - Y$ as spouse link in $\mathcal{G}$
25: **for each** $Z \in \left(\mathbf{Tri}(X-Y) \setminus \mathbf{S}_{XY}\right)$ **do**
26: $\mathbf{C} \leftarrow \mathbf{C} \cup \{(X \rightarrow Z \leftarrow Y)\}$
27: **end for**
28: **end if**
29: **end for**
30: remove all spouse links (i.e., marked links) from $\mathcal{G}$
31: **for each** orientation directive $(X \rightarrow Z \leftarrow Y) \in \mathbf{C}$ **do** /* orient edges */
32: **if** edges $X - Z$ and $Y - Z$ still exist in $\mathcal{G}$ **then**
33: orient edges as $X \rightarrow Z \leftarrow Y$
34: **end if**
35: **end for**
36: **return** $\mathcal{G}$
37: **end procedure**

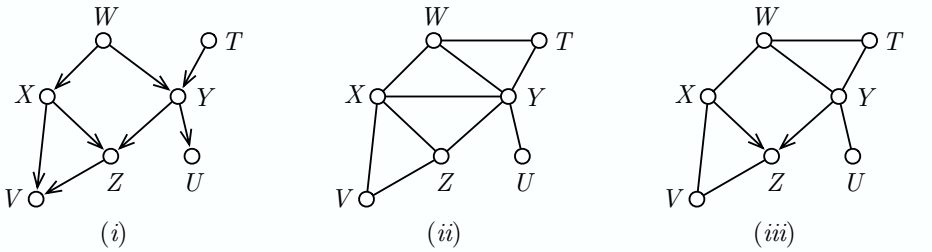

Figure 1: Sample local causal structure (i) and corresponding moral graph (ii). On (iii), the spouse link and orientation information that the collider set search for the linked pair $X-Y$ gives.

starting at line 5 in Algorithm 2. We are looking for a d-separating set for X and Y. Looking at the original graph, we see that $\{W\}$ is the smallest such set; let us see how the algorithm finds it. We have: $\mathbf{Tri}(X-Y)=\{W,Z\}$, $\mathbf{Bd}(X)=\{W,Y,Z,V\}$ and $\mathbf{Bd}(Y)=\{W,X,Z,U,T\}$. The base conditioning set $\mathbf{B}$ will thus be the smallest of $\big\{\{V\},\{U,T\}\big\}$, thus $\mathbf{B}=\{V\}$. At this stage, conditioning on V is justifiable: one cannot exclude situations where X and Y are d-connected given the empty set through T and V, for instance if T and V both had a common cause farther away in the network. But actually in this example, all (perfect) tests containing V in the conditioning set will yield dependence, because it is a descendant of the collider Z and thus opens a path by definition of d-separation. Eventually, in the iteration where $\mathbf{S}=\{W\}$, we will find conditional independence in the nested loop at lines 15 to 21. As $\mathbf{Tri}(X-Y)\setminus\mathbf{S}=\{Z\}$, $\mathbf{D}$ will be assigned the value $\{V\}$ and $\mathbf{B}'$ will be empty, so that we will perform exactly one extra test at line 17 with the conditioning set $\mathbf{S}_{XY}=\{W\}$, which yields independence. This in turn allows us to identify the link $X-Y$ as a spouse link and determine (line 25) that the set $\mathbf{Tri}(X-Y)\setminus\mathbf{S}_{XY}=\{Z\}$ is the set of all direct effects of X and Y; that is, fulfills the Collider Set property.

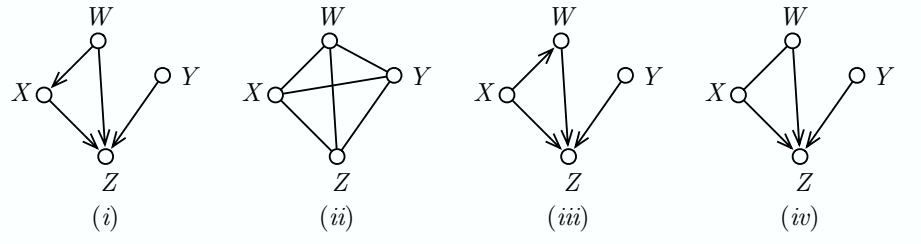

Figure 2: Another sample local causal structure (i) and corresponding moral graph (ii). On (iii), a wrong result if orientation is done immediately at line 26 of Algorithm 2. On (iv), the correct (non-)orientation if the condition at line 32 is added.

For some structures, the order in which arcs are removed and oriented must happen such that all spouse links are removed before proceeding to orientation. Consider another example, shown in Figure 2, and suppose again that that we are looking for a d-separating set for the pair (X,Y). As X and Y are unconditionally independent, $\mathbf{S}_{XY}=\emptyset$ is a valid d-separating set. We may thus remove the link $X-Y$, and considering that $\mathbf{Tri}(X-Y)=\{W,Z\}$, we could want to orient $X\rightarrow Z\leftarrow X$ and

$X \rightarrow W \leftarrow X$ (leaving the spouse link $W - Y$ to be removed later). This would be wrong, precisely because $W - Y$ is a spouse link, and thus the orientation $X \rightarrow W \leftarrow X$ is not allowed if one of the links to be oriented does not actually exist in the original graph. This is the reason why all orientation directives are saved in a list $\mathbf{C}$ at line 26 of Algorithm 2. After all spouse links have been removed, the orientations are done at line 33 only when both links to be oriented still exist, thus ensuring the existence of the V-structure $X \rightarrow Z \leftarrow Y$.

We do not claim that our algorithm uses the smallest possible conditioning set for the tests. There is a tradeoff between obtaining the minimal possible conditioning set and keeping the total number of tests low in the average case. In the empirical evaluation of this algorithm, we examine three behavioral criteria: the total number of tests, the average size of the conditioning set, and the maximum size of the conditioning set.

The complexity of the whole algorithm iterating over all triangle links, in terms of number of calls to CONDINDEP, is $O(d^2 2^{\alpha})$, where d is the number of variables and $\alpha = \max_{X \in \mathbf{V}} |\mathbf{Mb}(X)| - 1$. In the worst case of a fully connected graph, where $\mathbf{Mb}(X) = \mathbf{V} \setminus \{Y\}$, it is exponential in the number of variables due to the subset search. But in practice, the original graphs are often sparse enough so that the actual run time is not exponential. Many algorithms (e.g., MMMB, HITON_MB, AlgorithmMB, GS) perform subset searches in the (possibly augmented) Markov blanket and thus rely on graph sparseness to be efficient. Although the complexity of RESOLVEMARKOVBLANKETS_COLLIDERSETS is the same as that of RESOLVEMARKOVBLANKETS_GROWSHRINK, we show in the experimental results in Section 5 that the former performs fewer tests with a smaller average conditioning set size, while still providing comparable accuracy in structure learning.

3.3 A Generic Algorithm Based on Feature Selection

Thanks to the subroutine explained in the previous section, we can now draft a generic algorithm for structure learning based on feature-selection methods returning strongly relevant features. Algorithm 3 lists pseudocode for the three main steps of this approach:

1. Find the conjectured Markov blanket of each variable with feature selection and build the moral graph;
2. Remove spouse links and orient V-structures using collider sets;
3. Propagate orientation constraints.

For the sake of completeness, the constraint propagation rules of Step 3 have also been listed, in a separate subroutine (see Algorithm 4). They are common in structure learning to obtain a completed PDAG (Pearl and Verma, 1991). Meek (1995) proves that these three rules indeed return the maximally oriented graph when given a PDAG whose V-structures are oriented.

The challenge with this approach is twofold. One issue is efficiency: consistent but slow feature-selection algorithms will not beat existing causal learning algorithms, as they have to be run as many times as the number of variables d. The second and biggest issue is that consistent feature-selection algorithms are needed in order to prove correctness of this generic algorithm, in the sense that the result of feature selection should be equal to the set of strongly relevant features. This requirement is not always fulfilled. Hardin et al. (2004) study an SVM classifier and discuss feature selection based on the $\mathbf{w}$ weights: although irrelevant variables are not selected in the large sample limit, they show that the weights of the weakly relevant variables can be as close as one wishes to that of the strongly relevant variables due to the large-margin behavior of SVMs. Forward feature selection has been

Algorithm 3 Causal Structure Learning with Feature Selection

procedure GENERICSTRUCTURELEARNING
  Input: D : $n \times d$ data set with n d-dimensional data points
  Output: $\mathcal{G}$: maximally oriented partially directed acyclic graph

  /* Step 1: Markov blanket construction */
  for each variable $X \in \mathbf{V}$ **do**
    $\mathbf{F}_X \leftarrow$ FEATURESELECTIONALGORITHM(X, D)
  end for
  for each pair (X, Y) such that $Y \in \mathbf{F}_X$ and $X \in \mathbf{F}_Y$ **do** /* symmetry check */
    add X to $\mathbf{Mb}(Y)$ and Y to $\mathbf{Mb}(X)$
  end for

  /* Step 2: Spurious arc removal & V-structure detection */
  $\mathcal{G} \leftarrow$ RESOLVEMARKOVBLANKETS$(\mathbf{Mb}(\cdot))$

  /* Step 3: Constraint propagation */
  $\mathcal{G} \leftarrow$ COMPLETEPDAG$(\mathcal{G})$
  return $\mathcal{G}$
end procedure

Algorithm 4 Orient a PDAG maximally

procedure COMPLETEPDAG
  Input: $\mathcal{G}$: partially directed acyclic graph
  Output: $\mathcal{G}$: maximally oriented partially directed acyclic graph

  while $\mathcal{G}$ is changed by some rule **do** /* fixed-point iteration */
    for each X, Y, Z such that $X \rightarrow Y - Z$ **do**
      orient as $X \rightarrow Y \rightarrow Z$ /* no new V-structure */
    end for
    for each X, Y such that $X - Y$ and $\exists$ directed path from X to Y **do**
      orient as $X \rightarrow Y$ /* preserve acyclicity */
    end for
    for each X, Y s.t. $X - Y$ & $\exists$nonadj. Z, W s.t. $X - Z \rightarrow Y$ & $X - W \rightarrow Y$ **do**
      orient as $X \rightarrow Y$ /* three-fork V with married parents */
    end for
  end while
  return $\mathcal{G}$
end procedure

shown to miss strongly relevant variables (Guyon and Elisseeff, 2003). Nilsson et al. (2007) also describe forward selection as inconsistent, but claim that backward feature elimination is actually consistent in the large-sample limit.[9] For finite data sets, Statnikov et al. (2006) further show (among others) that even the weights of the irrelevant variables can get bigger than that of relevant variables, and that weakly relevant variables can be selected more often than strongly relevant variables in some cases.

These considerations are taken into account in our approach. In the next section, we describe an instantiation of the generic algorithm with an existing backward feature-elimination algorithm. Expecting the feature selection to be too inclusive, that is, to include features that are not strongly relevant, we add the filtering condition at line 5 of the generic outline in Algorithm 3: in order to link X and Y in the moral graph, we require the feature selection performed for X to have selected variable Y, and conversely, we require X to have been selected by the feature selection performed for Y. This does not theoretically guarantee the absence of "false positives," however. Further in the section, we replace the feature-selection step with a provably consistent algorithm in the multivariate Gaussian case, and analyze its complexity and behavior.

4. Algorithms for Causal Feature Selection

In this section, we show two algorithms (and a variant) as instantiations of the generic approach previously described. First, we explain an algorithm based on the Recursive Feature Elimination (RFE) algorithm (Guyon et al., 2002) as a direct application of existing methods. We then describe Total Conditioning (TC), a fast algorithm that can be proved correct under the multivariate Gaussian assumption. We also show a variant, TC_{bw}, that improves accuracy with low sample sizes by using an explicit backward feature-selection heuristics. In Section 5, we report on experiments including these algorithms.

4.1 An RFE-Based Approach

To empirically test the soundness of the approach, we propose to use RFE over a Support Vector Regression (SVR) learner (Smola and Schölkopf, 1998) with a linear kernel, assuming for this example that we will deal with multivariate Gaussian data. RFE is an instance of a backward feature-elimination algorithm. Given some learner (in this case, SVR), it iteratively trains it, ranks the features according to some criterion, and remove the feature (or the p features) with the smallest ranking criterion. This criterion can be the weights $\mathbf{w}$ attributed to the features by the learner, or some sensitivity measure of the features (Guyon et al., 2002). In our case, we used the weights $\mathbf{w}$ of SVR as described in Smola and Schölkopf (1998).

Using RFE, the Markov blanket identification is done in two steps:

1. Use RFE to rank the predictors according to their weights in the trained model and to provide what can be seen as a relevance ordering of the predictors;

2. Determine the size of the Markov blanket and thus the number of variables to select from the list returned by RFE.

9. This is subject to the assumption that the underlying classifier must itself be consistent, in the sense that it must return the Bayes classifier in the large-sample limit.

We do not have a theoretical guarantee that RFE/SVR will return the Markov blanket variables. Although Nilsson et al. (2007) shows that RFE/SVM as described in Guyon et al. (2002) is consistent (i.e., returns strongly relevant variables in the large-sample limit), the limitations of ranking variables on the **w** weights of an SVM with finite data sets have also been highlighted (Hardin et al., 2004; Statnikov et al., 2006). For now, we thus use this feature-selection step as a heuristics.

In order to determine the number of variables to select from the ranked list returned by RFE, we use the following criterion: starting with the first variable from the list, accept a new variable in the Markov blanket if the cross-validated training error of the SVR decreases with the new variable, and stop and return the current list if adding the next variable increases the error.

Algorithm 5 An RFE-Based Feature-Selection Step

1: **procedure** RFEFEATURESELECTION
   Input: X : the target variable to perform feature selection for
   D : $n \times d$ data set with n d-dimensional data points
   Output: $\mathbf{S}$: the set of selected variables
2: $\mathbf{w} \leftarrow$ weights of $\mathbf{V} \setminus X$ according to RFE(SVR)
3: $\mathbf{P} \leftarrow$ predictor variables sorted according to $\mathbf{w}$
4: $\mathbf{S} \leftarrow \emptyset$
5: $error_{opt} \leftarrow \text{var}[X]$ /* MSE of constant function */
6: $error \leftarrow$ TRAIN(cross-validated SVR with predictor $(\mathbf{P})_1$))
7: **while** $error < error_{opt}$ **do**
8: $error_{opt} \leftarrow error$
9: $\mathbf{S} \leftarrow \mathbf{S} \cup \{ (\mathbf{P})_1 \}$ /* add beneficial predictor */
10: $\mathbf{P} \leftarrow \mathbf{P} \setminus \{ (\mathbf{P})_1 \}$
11: $error \leftarrow$ TRAIN(cross-validated SVR with predictors $\mathbf{S} \cup \{ (\mathbf{P})_1 \}$)
12: **end while**
13: **return S**
14: **end procedure**

The symmetry condition (2), $X \in \mathbf{F}_Y \Leftrightarrow Y \in \mathbf{F}_X$, might not be satisfied: we rely on the check at line 5 of the generic approach of Algorithm 3 to make sure that we do not select spurious features in the Markov blanket. This conservative approach implies that we expect RFE to select at least all strongly relevant variables, plus possibly some others that we hope to identify with this simple condition.

As a conditional-independence test at lines 9 and 17 of the collider set search in Algorithm 2, we can use the distribution-free Recursive Median (RM) algorithm proposed by Margaritis (2005) to detect the V-structure and remove the spouse links, or a z-test as used in Scheines et al. (1995) in the case of Gaussian data.

Although we expect the resulting graph to be accurate in the large sample limit (see Section 5), we also expect the run time of such an approach to be much higher compared to existing algorithms. Training the SVR has a cubic complexity in terms of the number of samples, $O(n^3)$. To get an accurate ranking, RFE runs the training $d-1$ times. Then, a new SVR learner is trained and cross-validated several times (we used a 5-fold cross-validation) to get the validation error, which is repeated for each variable in the actual Markov blanket. The complexity for the whole feature-selection step is then $O(d^2 n^3)$, with a large constant factor. We thus emphasize that this RFE-based

feature selection is not meant as a valid practical instantiation of the generic algorithm, but rather as a proof of concept to validate the approach. In order to be practical, the feature-selection step has to be redesigned so that it is done efficiently when run for all variables. This is what the next algorithm is meant to address in the specific case of multivariate Gaussian variables.

4.2 The TC Algorithm

We now propose in the procedure TCFEATURESELECTION (Algorithm 6) another instantiation of the feature-selection call at line 3 of the generic approach of Algorithm 3. The whole algorithm as determined by the feature-selection, collider-identification, and maximal-orientation steps is equivalent to the TC algorithm described in Pellet and Elisseeff (2007). (We thus write "TC" to refer to the whole algorithm and not only to the feature-selection procedure, referred to as TCFEATURESELECTION.)

For a given target variable X, TC estimates the coefficients of a multiple regression problem, considering all other variables $\mathbf{V} \setminus X$ as predictors. It then returns the significant predictors, according to a t-test on the coefficient of each variable. Its short listing is in Algorithm 6.

Algorithm 6 The Total Conditioning Feature-Selection Step

1: **procedure** TCFEATURESELECTION
   Input: X : the target variable to perform feature selection for
   D : $n \times d$ data set with n d-dimensional data points
   Output: $\mathbf{S}$: the set of selected variables
2: $\mathbf{b} \leftarrow$ weights of $\mathbf{V} \setminus X$ in the problem of regressing X on $\mathbf{V} \setminus X$
3: $\mathbf{S} \leftarrow \{$predictors whose b weight is significant$\}$
4: **return S**
5: **end procedure**

The conditional-independence tests to be performed at lines 9 and 17 of the collider set search of Algorithm 2 are done using partial correlation.

Definition 11 (Partial correlation) *In a variable set* $\mathbf{V}$*, the* partial correlation *between two random variables* $X, Y \in \mathbf{V}$ *given* $\mathbf{Z} \subseteq \mathbf{V} \setminus \{X, Y\}$*, noted* $\rho_{XY \cdot \mathbf{Z}}$*, is the correlation of the residuals* R_X *and* R_Y *resulting from the least-squares linear regression of* X *on* $\mathbf{Z}$ *and of* Y *on* $\mathbf{Z}$*, respectively.*

TC was shown to be correct in the large sample limit (subject to the consistency of the statistical tests) in Pellet and Elisseeff (2007) under the Faithfulness and Causal Sufficiency assumptions. For the sake of completeness, we add the proof to Appendix A. The main points leading to the correctness of TC are the equivalence of a zero regression weight for some predictor Y while regressing X on all variables $\mathbf{V} \setminus X$ and a zero partial correlation $\rho_{XY \cdot \mathbf{V} \setminus \{X,Y\}}$, and the fact that this is zero if and only if $(X \perp\!\!\!\perp Y \mid \mathbf{V} \setminus \{X, Y\})$ holds in a Gaussian context (Baba et al., 2004). Then, our feature-selection step (Algorithm 6) gives the Markov blanket for each node, and the collider set search (Algorithm 2) then takes care of identifying the V-structures and removing the spouse links.

The other advantage of using linear regression and partial correlation is that it yields a fast algorithm. Actually, *all* regression weights and parameters needed for the feature-selection step of TC can be efficiently computed by inverting the sample correlation matrix $\mathbf{R} \in [-1, 1]^{d \times d}$. Building

graphs by inverting the correlation matrix is typically what is done with Gaussian Markov random fields, a special case of undirected graphical models (see, e.g., Talih, 2003).

The weight computation and the statistical significance tests are performed as follows. Let $\hat{b}_{ij}$ be the maximum likelihood estimator of the true regression weight b_{ij} of predictor X_j when X_i is the dependent variable, such that it solves the multiple regression equation for target X_i in the sense that it minimizes the sum of the squared residuals

$$SS_R = \sum_{k=1}^{n} \left(x_{ik} - \sum_{j=1, j\neq i}^{d} \hat{b}_{ij} x_{jk} \right)^2$$

where x_{ik} is the value of X_i for the kth sample. If we have the inverse correlation matrix $\mathbf{R}^{-1} = (r^{ij})$, the vector $\mathbf{b}$ at line 2 of Algorithm 7 can be found in linear time: $\hat{b}_{ij} = -r^{ij}/r^{ii}$ (Raveh, 1985). For instance, the list of weights to predict variable X_1 with all others is

$$\mathbf{b}_1 = (\hat{b}_{12}, \hat{b}_{13}, \cdots, \hat{b}_{1d}) = -(r^{12}, r^{13}, \cdots, r^{1d})/r^{11}. \tag{9}$$

The distribution of these weights is known (Judge et al., 1988):

$$\frac{\hat{b}_{ij} - b_{ij}}{\hat{\sigma}_{ij}} \sim t_{(n-(d-1))}, \tag{10}$$

where $\hat{\sigma}_{ij}$ is the standard error of the jth predictor for variable X_i; that is, that it follows a t distribution with a number of degrees of freedom $df =$ number of samples $-$ number of predictors $= n-(d-1)$. For our null hypothesis $H_0 : b_{ij} = 0$, we need $\hat{\sigma}_{ij}$ in addition to $\hat{b}_{ij}$ to compute the t-statistics $\hat{b}_{ij}/\hat{\sigma}_{ij}$. The estimate of the coefficient error $\hat{\sigma}_{ij}$ can be expressed as

$$\hat{\sigma}_{ij} = \hat{\sigma}_i \sqrt{\omega^{jj}/n},$$

where $\hat{\sigma}_i$ is an estimator of the standard error of the regression for target X_i, and ω^{jj} is the jth diagonal element of the inverse correlation matrix of the predictors (Judge et al., 1988, p. 243). (How to obtain the inverse correlation matrix of the predictors from the $\mathbf{R}^{-1}$ matrix in quadratic time is discussed in the next subsection.) The standard error $\hat{\sigma}_i$ can also be obtained in linear time from $\mathbf{R}^{-1}$ as follows.

Without loss of generality, we assume a zero mean and a unit standard deviation for all variables. Then $\sigma_i^2 = 1 - R_i^2$, where R_i^2 is the coefficient of determination of the regression for target X_i. This coefficient can be expressed as the scalar product of the $\mathbf{b}_i$ vector with the vector $\mathbf{r}_i$ of the pairwise correlation coefficients of the predictors with the target X_i (Raveh, 1985), which we read directly from the correlation matrix $\mathbf{R}$:

$$R_i^2 = \mathbf{b}_i^T \mathbf{r}_i.$$

An unbiased estimator $\hat{\sigma}_i$ for σ_i is then

$$\hat{\sigma}_i = \sqrt{\frac{n(1 - \mathbf{b}_i^T \mathbf{r}_i)}{n-d}}.$$

To sum up, we have a complexity of $O(nd^3)$ to build and invert the correlation matrix, and $O(d^3)$ to check for significance. This comes from having to obtain d times the inverse correlation matrix of $d-1$ predictors in $O(d^2)$, and then checking their significance in linear time. The overall complexity of TC, including the collider identification and the constraint-propagation steps, is then $O(nd^3 + d^2 2^{\alpha})$.

The weaknesses of this approach are its infeasability when the correlation matrix $\mathbf{R}$ does not have full rank (including the special case $n < d$, that is, when there are fewer samples than variables), the low power of the statistical tests with small data sets, and multicollinearity in the predictors. The symptoms of the last two points are that the t-tests do not refute the null hypothesis of zero weight because (i) there is not enough data to support it, or (ii) multicollinearity makes the weights lower than they should be, such that it becomes harder to interpret them as depicting the independent contribution of each predictor. We try to deal with this problem in the next section with the TC_{bw} algorithm.

4.2.1 SIGNIFICANCE TESTS

Independently of low sample sizes or multicollinearity, the statistical tests on the weights of the linear regression equations are a delicate point in TC. The choice of the Type I error rate α needs investigating as it significantly influences the result of the algorithm.

In a network of d nodes, the feature-selection step performs $d(d-1)/2$ tests to determine the undirected skeleton. We will falsely reject the null hypothesis $b_{ij} = 0$ about $m \cdot \alpha$ times on average, where $m < d(d-1)/2$ is the difference in the number of edges between the original DAG $\mathcal{G}_0$ and the complete graph. We will thus add on average $m \cdot \alpha$ wrong edges. We can set the significance level for the individual tests to be inversely proportional to $d(d-1)/2$ to avoid this problem (assuming a large m and thus rather sparse graphs), and check that it does not affect the Type II error rate too much, which we do now.

According to (10), the expression $(\hat{b}_{ij} - b_{ij})/\hat{\sigma}_{ij}$ follows a t distribution with $n-(d-1)$ degrees of freedom. If we call $\Psi(\cdot)$ the cumulative distribution function of a t distribution with $n-(d-1)$ degrees of freedom, we can write the Type II error rate β for each regression weight:

$$\beta_{ij} = \Psi(\Psi^{-1}(1-\alpha/2) - |b_{ij}|/\hat{\sigma}_{ij}).$$

The values for $\hat{\sigma}_{ij}$ can be computed from the inverse correlation matrix $\mathbf{R}^{-1}$ and thus depend on the particular data set being analyzed, but the true b_{ij} are unknown. What we could do in theory to optimize α is to minimize the average number of extraneous (T_e) and missing (T_m) links:

$$T = T_e + T_m = m \cdot \alpha + \sum_{(i,j) \in \mathbf{E}} \beta_{ij},$$

where m is the number of edges missing in the original DAG compared to a full graph, and $\mathbf{E}$ is the set of arcs in the original DAG, so that $m + |\mathbf{E}| = d(d-1)/2$. As m, $\mathbf{E}$ and b_{ij} are unknown, we can only find an upper bound for the number of missed links T_m, provided (i) we can estimate the graph sparseness to approximate m; (ii) we assume $|b_{ij}| \geq \delta$; and (iii) we choose $\mathbf{E}^{\star}$ such that it maximizes the sum in (11), with $|\mathbf{E}^{\star}| = d(d-1)/2 - m$. Then we have:

$$T_m \leq \sum_{(i,j) \in \mathbf{E}^{\star}} \Psi(\Psi^{-1}(1-\alpha/2) - \delta/\hat{\sigma}_{ij}). \tag{11}$$

Although this bound was found too loose for practical use, we can model the Type I and Type II error rate as a function of α for artificial problems whose sparseness and regression weights are known. This is shown in Figure 3 for a specific instance of an Alarm data set (see Section 5 for details on this network) with two different sample sizes, $n = 50$ (left) and $n = 250$ (right). We did not use this information to tune α in the experiments, as it cannot be obtained without prior knowledge, but the curves showed that an α inversely proportional to $d(d-1)/2$ has the same order of magnitude as the optimal α on the data sets we analyzed.

What we also see is that the Type I error curve rapidly goes up, whereas the Type II error curve is upper-bounded by the total number of links in the original graph. In terms of pure number of errors, setting a low α will thus be more beneficial than setting a higher α to get a low β. It is worth discussing, however, depending on the particular problem to solve, which is more desirable: missing causal links or getting extra causal links. In terms of Bayesian networks, getting too few links prevents the model from being able to reconstruct the full joint probability distribution, because we lose the I-map property; whereas getting too many links implies having to estimate more parameters from the same data and thus complexifies a subsequent parameter learning task.

4.3 The TC_{bw} Algorithm

Despite correctness of TC, with a low number of samples n it fails to have enough evidence for rejecting the null hypothesis of zero regression weight, and thus misses links (see detailed results in Section 5), even for a high α. We now try to address this particular issue by successively eliminating the most insignificant predictors and reevaluating the remaining ones. This is actually a backward stepwise-regression method. Pseudocode for this heuristics is listed in Algorithm 7.

Algorithm 7 The Total Conditioning Backward Feature-Selection Step

```
1:  procedure TCBWFEATURESELECTION
      Input:   X :  the target variable to perform feature selection for
               D :  n × d data set with n d-dimensional data points
      Output:  S :  the set of selected variables
2:    P ← V \ X    /* all predictors */
3:    S ← ∅    /* significant predictors */
4:    while P ≠ ∅ and P ≠ S do
5:      b ← weights of P in the problem of regressing X on P
6:      S ← S ∪ {predictors whose b weight is significant}
7:      P ← P \ {the p less significant predictors}
8:    end while
9:    return S
10: end procedure
```

Intuitively, the problem to solve is that the regression weights cannot be high enough for significance with small sample sizes. By removing the most insignificant predictors and thus the most likely to be actually zero, we scale down the regression problem and increase the power of the tests. How many insignificant predictors to remove can be discussed; in our implementation, we compared $p = 1$ to $p = (\text{number of predictors})/2$ and found that the latter yielded results that were just as good with an important speed gain.

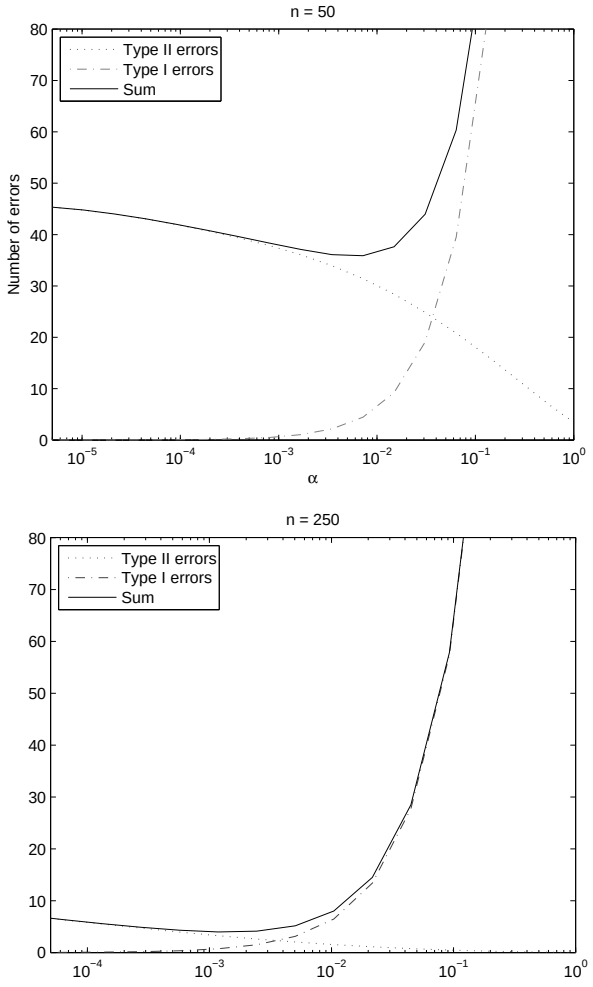

Figure 3: Expected Type I and II errors as a function of α

This stepwise regression raises some issues; notably, Tibshirani (1994) argues that the repeated tests on non-changing data are biased and that the remaining **b** coefficients are too large. We thus expect TC_{bw} to be biased and to include more false positives than TC. Ideally, one would need a criterion to predict when the additional false positives would outweigh the benefits of reducing the false negatives. Whether such a criterion, which would allow us to know a priori whether TC or TC_{bw} should be used, can be found, is an open question.

Solving a standard multiple regression problem with d predictors traditionally has complexity $O(nd^3)$. Naïvely solving $d-1$ regression problems d times in the case $p=1$ would have a complexity of $O(nd^5)$. But we can avoid reinverting matrices in the inner loop of the stepwise regression thanks to the following result.

Let $\Sigma = \mathbf{X}^T\mathbf{X}$ be n times the correlation matrix $\mathbf{R}$, where $\mathbf{X}$ is the $n \times d$ matrix representing a data set where all variables have zero mean and unit standard deviation. Then we can use Σ^{-1} to linearly find the weights of the regression problems and their standard error, which are needed for

the t-tests. Suppose we find that variable X_1 is the weakest predictor, and want to reevaluate the weights of the other predictors at line 5 of TC_{bw}. Let $\mathbf{X}_{\setminus i}$ be the data set where variable X_i has been removed. Then we need the matrix Ω^{-1} to solve the new problem, where $\Omega = \mathbf{X}_{\setminus 1}^T \mathbf{X}_{\setminus 1}$. As a special case of Strassen's blockwise matrix inversion formula, we have:

$$\Sigma = \begin{bmatrix} \sigma_{11} & \mathbf{c}^T \\ \mathbf{c} & \Omega \end{bmatrix}$$

$$\Longrightarrow \quad \Sigma^{-1} = \begin{bmatrix} \dfrac{1}{\sigma_{11} - \mathbf{c}^T \Omega^{-1} \mathbf{c}} & -\dfrac{\mathbf{c}^T \Omega^{-1}}{\sigma_{11} - \mathbf{c}^T \Omega^{-1} \mathbf{c}} \\ -\dfrac{\Omega^{-1} \mathbf{c}}{\sigma_{11} - \mathbf{c}^T \Omega^{-1} \mathbf{c}} & \Omega^{-1} + \dfrac{\Omega^{-1} \mathbf{c} \mathbf{c}^T \Omega^{-1}}{\sigma_{11} - \mathbf{c}^T \Omega^{-1} \mathbf{c}} \end{bmatrix}.$$

Let $\sigma^{ij} = (\Sigma^{-1})_{ij}$ and $\mathbf{b} = \Omega^{-1}\mathbf{c}$. Then $\mathbf{b}$ are the weights of the regression of X_1 on $X_2, \cdots, X_d$ and can be computed without knowing Ω^{-1} (Raveh, 1985), see (9). We have:

$$\sigma^{11} = 1/(\sigma_{11} - \mathbf{c}^T \mathbf{b})$$

and, $(\Sigma^{-1})_{\setminus 1}$ being the matrix Σ^{-1} where the first row and column have been removed,

$$(\Sigma^{-1})_{\setminus 1} = \Omega^{-1} + \mathbf{b}\mathbf{b}^T / (\sigma_{11} - \mathbf{c}^T \mathbf{b}).$$

We can thus compute Ω^{-1} given Σ^{-1} with complexity $O(d^2)$ as follows:

$$\Omega^{-1} = (\Sigma^{-1})_{\setminus 1} - \sigma^{11} \mathbf{b}\mathbf{b}^T. \tag{12}$$

This trick is also used in TC to find the inverse correlation matrix of the predictors from the inverse correlation matrix of the whole variable set.

Equation (12) is implemented in TC_{bw} such that we never need to invert another matrix again once Σ^{-1} has been obtained, and leads to a complexity of $O(d^2)$ for stepwise elimination of a predictor. In the most computationally expensive case $p = 1$, this elimination of row and column of the inverse matrix is repeated at most $d - 2$ for each variable, yielding a complexity of $O(nd^4)$ for the whole feature-selection step for all variables. The overall complexity of TC_{bw} is then $O(nd^4 + d^2 2^{\alpha})$. We are only adding one complexity degree in d with respect to TC with the additional stepwise regression.

5. Experimental Results

In this section, we report on experiments and results on two points separately. First, we test our procedure described in Algorithm 2 to recover the local structure with the collider set search given all Markov blankets, and compare it to the relevant steps of the GS algorithm, which are listed in Algorithm 1, with 5 different network topologies. For the sake of comparison, we also run the reference PC algorithm (Spirtes et al., 2001), initialized with the moral graph instead of the fully connected graph.

Second, we conduct experiments to investigate how the whole structure-learning algorithms behave. We first use the RFE-based approach. We then systematically compare TC, TC_{bw} and several reference algorithms, varying the data set size and the network size. Note that results for some algorithms may be sparser due to their prohibitive run times on some data sets.

5.1 Experimental Setup

In order to test the accuracy of the various algorithms, we chose to sample data from the following known networks, from the Bayes net repository (Elidan, 2001):

- Alarm network (Beinlich et al., 1989). This network has become a de facto standard benchmark for structure-learning algorithms: it contains 37 nodes, 46 arcs, 4 undirected in the PDAG of the equivalence class. It was originally designed to help interpret monitoring data to alert anesthesiologists to various situations in the operating room. It is depicted in Figure 4.

- Insurance (Binder et al., 1997), 27 nodes, 52 arcs, 18 undirected in its PDAG. It was designed to evaluate car insurance risks. This network has fewer nodes than Alarm but is denser, see Figure 5.

- Hailfinder (Abramson et al., 1996), 56 nodes, 66 arcs, 17 undirected in its PDAG. It is a normative system that forecasts severe summer hail in northeastern Colorado. See Figure 6.

- Carpo,[10] 61 nodes, 74 arcs, 24 undirected in its PDAG. It is meant to help diagnose the carpal tunnel syndrome. The version we used has three disconnected subgraphs, one of which is a single variable, and a relatively flat causal structure, as can be seen in Figure 7.

- A subset of Diabetes (Andreassen et al., 1991) with 104 nodes, 149 arcs, 8 undirected in its PDAG, which was designed as a preliminary model for insulin dose adjustment. This subset is made of 6 repeating patterns (there are 24 in the original network) of 17 nodes, plus 2 external nodes linked to every pattern. The first two of these patterns are shown in Figure 8.

We performed three series of experiments.

1. We compared our algorithm resolving the Markov blanket to the relevant steps of the Grow-Shrink algorithm, as described in Section 3.2, and to PC;

2. We tested the RFE-based approach and compared it to PC;

3. Finally, we compared TC and TC_{bw} to three reference algorithms and examine their accuracy, run time, and number of tests while varying the network structure, the network size, and the sample size.

The chosen reference algorithms are:

1. The PC algorithm. PC is, like TC and TC_{bw}, exponential in the worst case, when graphs are not sparse enough: we discuss which structural elements make PC or TC exhibit the exponential behavior;

2. The full Grow-Shrink algorithm, as described in Margaritis and Thrun (1999);

3. A state-of-the-art Bayesian structure-learning algorithm that works with continuous data sets, the Bach-Jordan scoring algorithm (Bach and Jordan, 2003), coupled with a greedy search in the space of DAGs. Note that Bayesian structure-learning algorithms are often score-based and return fully oriented DAGs. Maximizing the chosen score function might not minimize the number of structural errors as we report in these results.

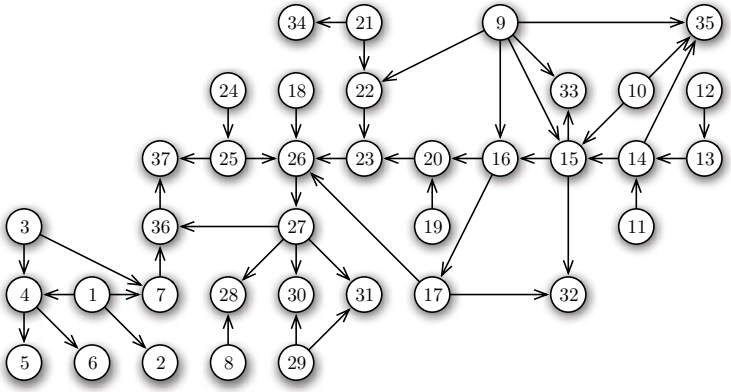

Figure 4: The Alarm network

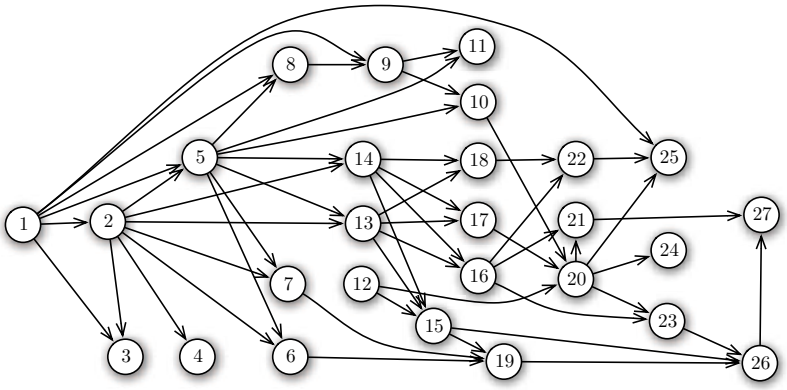

Figure 5: The Insurance network

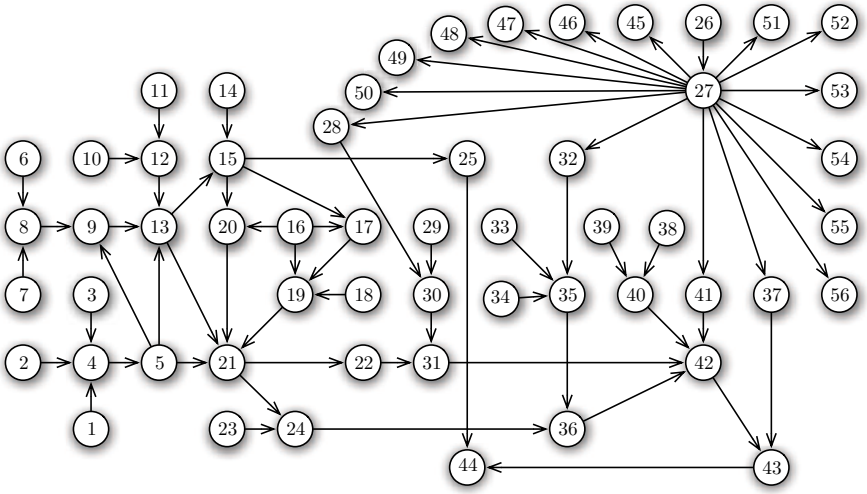

Figure 6: The Hailfinder network

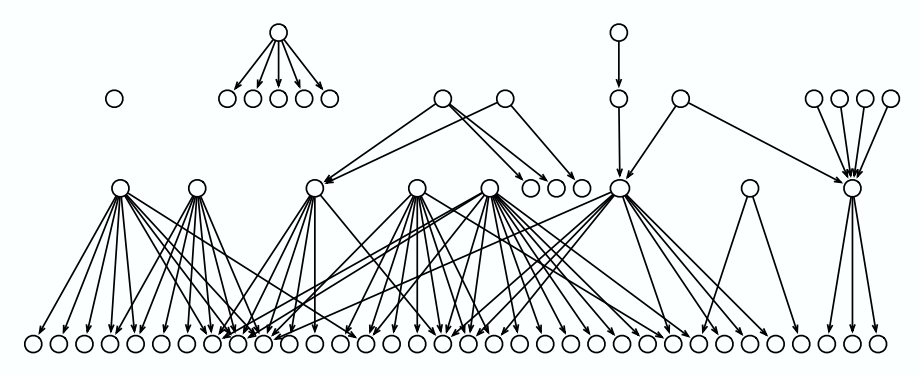

Figure 7: The Carpo network

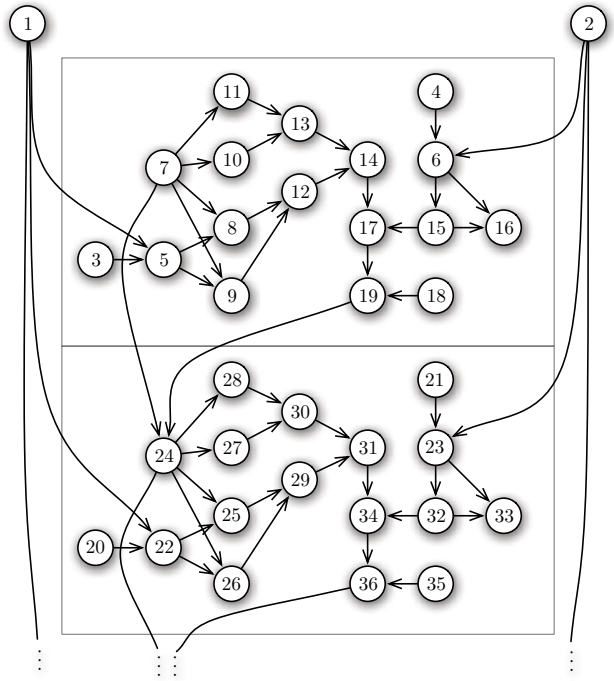

Figure 8: Two of the six patterns of the Diabetes network

For all simulation experiments, we generated the data sets by using the 5 graphs as a structure for a linear structural equations model: the parentless variables were sampled as Gaussians with zero mean and unit standard deviation; the other variables were defined as a linear combination of their parents with coefficients randomly distributed uniformly between 0.2 and 1, similarly to what was done in Scheines et al. (1995) for the evaluation of PC. The disturbance terms were also normally distributed. We compared the number of tests, the size of the conditioning sets, and the

10. Created by Alex Dagum with contributions from Mark Peot, as indicated on its page at the Bayes net repository. No corresponding publication was found.

structural errors in case of runs with artificial data. A structural error is an arc addition, deletion, or reversal with respect to the original graph.

We used the implementation of PC proposed by Leray and François (2004) in the BNT Structure Learning Matlab package. The implementation of TC and TC_{bw} was also done in Matlab. The statistical tests were done using Fisher's z-transform of the partial correlation, unless otherwise stated. For PC and GS, we chose the default value of $\alpha = 0.05$; we note though that the optimal value of α is problem dependent and that especially with low sample sizes, hand tuning α can return better results than those listed here. For both TC and TC_{bw}, we set $\alpha = 2/(d(d-1))$, according to the discussion at the end of Section 4.2.

5.2 Local Structure Recovery with Markov Blanket Information

In this series of experiments, we compare RESOLVEMARKOVBLANKETS_COLLIDERSETS (CS) to RESOLVEMARKOVBLANKETS_GROWSHRINK and to a modified version of PC, where the graph being built is initialized with the moral graph (instead of the full graph in the original version of PC). This represents exactly the Markov blanket information available to the two other algorithms and allows a direct comparison. Note that we observe the PDAG that PC obtains *before* the constraint-propagation step building the maximally oriented PDAG, such that, in all three tested algorithms, we only expect the V-structures to be oriented.

We tested the three algorithms on each network using two methods to check for conditional independence: first, using a d-separation oracle with the original graph (which is equivalent to a perfect test); and second, using Fisher's z-transform of the sample partial correlation coefficient as computed on artificial data, with significance $\alpha = 0.05$. Using the oracle always yields correct graphs.

Table 1 shows the results of these experiments. We first list the results obtained when using a d-separation oracle to decide upon conditional independence. For GS, we ran two versions of Algorithm 1: one, which we name GS(1), where the subset searches at lines 5 and 13 proceed with decreasing sizes of the chosen subset $\mathbf{S}$, and another, GS(2), with increasing subset sizes. GS(1) usually leads to fewer tests, but with larger conditioning sets. The order of the subset searches for our method (lines 7 and 15 in Algorithm 2) was fixed to decreasing subset sizes, as this always led to fewer tests *and* smaller conditioning sets.

The results for the modified PC algorithm are only shown for the sake of comparison: PC is a general-purpose algorithm which is not specialized in such local structure recognition given the Markov blankets. What the comparison shows, however, is that, whenever this Markov blanket information is available or cheap to obtain, there are much more efficient approaches.

GS(1) and GS(2) are close to one another in all scores, and outperform PC (by several orders of magnitude) in the number of tests and (significantly) in average and maximum size of the conditioning sets (except, artificially, for the results marked with a star), because it uses the Markov blanket information better. Our approach, however, is one order of magnitude better than GS(1) and GS(2) in terms of number of tests, while still using smaller average and maximum conditioning set sizes in all tested networks. Especially striking are the results on the Carpo network: this is an example where CS saves a lot of time ignoring the numerous links not part of triangles, whereas GS(1) and GS(2) also checks those, with the often large Markov blankets (Figure 7).

We then performed the same experiments, but using the statistical tests on data sampled from the networks as described in the previous sections. We used a fixed sample size $n = 500$ and averaged

Algorithm	Alarm	Insurance	Hailfinder	Carpo	Diabetes
modified PC					
# tests	11331	773572	19543985*	2025250*	93134*
avg. $\lvert\mathbf{Z}\rvert$	4.36	7.65	5.75*	5.47*	4.64*
max $\lvert\mathbf{Z}\rvert$	10	16	6*	6*	6*
GS(1)					
# tests	1485	6435	2809	209342	5414
avg. $\lvert\mathbf{Z}\rvert$	2.62	3.63	2.66	7.46	2.73
max $\lvert\mathbf{Z}\rvert$	8	11	7	15	10
GS(2)					
# tests	1472	7180	2979	200621	6197
avg. $\lvert\mathbf{Z}\rvert$	2.20	3.05	2.31	7.39	2.39
max $\lvert\mathbf{Z}\rvert$	7	8	7	15	8
CS					
# tests	214	1288	593	294	943
avg. $\lvert\mathbf{Z}\rvert$	1.80	2.69	2.30	1.79	2.13
max $\lvert\mathbf{Z}\rvert$	5	6	6	8	7

Table 1: Number of tests and size of the conditioning sets (noted $\lvert\mathbf{Z}\rvert$) as performed by various algorithms to recover the local network structure of the networks given perfect Markov blanket information. The star (*) notes PC results where the maximum size of the conditioning set has been set to 6 to avoid prohibitive run times.

over 9 different samplings for each network. We only compared PC, GS(1) and CS on this series of experiments, preferring GS(1) to GS(2) because of the lower number of tests it usually performs. The exhaustive results are listed in Table 2 for the sake of completeness, and the sum of the structural errors is also shown in Figure 9 for easier visualization.

First, we see that we get similar results as in Table 1 as far as the number of tests and size of the conditioning sets are concerned: CS is faster and consistently performs fewer tests with smaller conditioning sets, which leads to an increased power of the tests. However, that is sometimes balanced out by the fact that CS relies on a single series of tests both to remove spouse links and to orient (possibly multiple) V-structures at the same time, thus leading to a greater penalty if the outcome of a test is wrong with respect to the initial graph.

We see that GS(1) and PC can beat CS on certain arc scores; PC, in particular, is good at avoiding arc orientation mistakes in these experiments. GS(1), which checks not only triangle links but all links to try to orient them, makes more orientation mistakes, especially on the Carpo network. PC tends to miss a few more arcs than CS, which in turn misses a few more than GS(1). But in total, CS beats GS(1) significantly on Insurance, Hailfinder, and Carpo, while performing slightly better on Alarm and being slightly outperformed on Diabetes. Based on these results, we will now use our collider set search as the method of choice to break up the Markov blankets for the next series of experiments.

Algorithm	Alarm	Insurance	Hailfinder	Carpo	Diabetes
mod. PC					
# tests	2850 ± 285	13461 ± 3247	$9681105^{\dagger}$	412791 ± 104080	57153 ± 9910
avg. $\lvert\mathbf{Z}\rvert$	2.97 ± 0.17	3.50 ± 0.33	$5.54^{\dagger}$	5.17 ± 0.15	4.37 ± 0.14
max $\lvert\mathbf{Z}\rvert$	6	6	$6^{\dagger}$	6	6
arcs:					
missing	5.44 ± 0.53	9.56 ± 1.01	$6^{\dagger}$	14.22 ± 1.64	9.56 ± 1.88
extra	0.33 ± 0.5	0.11 ± 0.33	$\mathbf{0}^{\dagger}$	$\mathbf{0.22} \pm 0.44$	$\mathbf{1.11} \pm 0.60$
reversed	**0**	$\mathbf{0.22} \pm 0.67$	$1^{\dagger}$	0.11 ± 0.22	$\mathbf{2.00} \pm 1.39$
TOTAL	5.78 ± 0.72	9.89 ± 1.47	$7^{\dagger}$	14.56 ± 1.86	12.67 ± 2.69
GS(1)					
# tests	1304 ± 60	4544 ± 195	2415 ± 63	129265 ± 17033	5239 ± 46
avg. $\lvert\mathbf{Z}\rvert$	2.66 ± 0.10	3.66 ± 0.04	2.62 ± 0.02	7.49 ± 0.08	2.76 ± 0.01
max $\lvert\mathbf{Z}\rvert$	8	11	7.89 ± 0.33	15	10
arcs:					
missing	$\mathbf{1.56} \pm 0.53$	$\mathbf{5.44} \pm 0.53$	$\mathbf{3.11} \pm 0.33$	**0**	6.11 ± 0.78
extra	0.56 ± 0.73	0.33 ± 0.71	1 ± 0.71	$\mathbf{0.22} \pm 0.44$	2.78 ± 1.64
reversed	1.11 ± 1.05	3.67 ± 2.12	8 ± 2.29	16.78 ± 2.49	2.67 ± 2.00
TOTAL	3.22 ± 1.81	9.44 ± 2.39	12.11 ± 2.74	17 ± 2.62	$\mathbf{11.55} \pm 3.03$
CS					
# tests	173 ± 3	782 ± 19	507 ± 18	308 ± 14.39	907 ± 4
avg. $\lvert\mathbf{Z}\rvert$	1.55 ± 0.03	2.36 ± 0.02	2.08 ± 0.03	1.90 ± 0.12	2.17 ± 0.01
max $\lvert\mathbf{Z}\rvert$	5	6	5	8	7
arcs:					
missing	$\mathbf{1.56} \pm 0.73$	6.33 ± 0.5	3.44 ± 0.52	**0**	$\mathbf{5.11} \pm 1.17$
extra	$\mathbf{0.44} \pm 0.53$	$\mathbf{0.22} \pm 0.44$	0.67 ± 0.70	0.33 ± 0.50	1.44 ± 1.51
reversed	0.11 ± 0.33	0.33 ± 0.5	$\mathbf{0.11} \pm 0.33$	**0**	7.11 ± 1.05
TOTAL	$\mathbf{2.11} \pm 0.96$	$\mathbf{6.89} \pm 1.03$	$\mathbf{4.22} \pm 1.27$	$\mathbf{0.33} \pm 0.50$	13.66 ± 2.20

Table 2: Number of tests, size of the conditioning sets (noted $\lvert\mathbf{Z}\rvert$), and structural errors as returned by GS(1) and CS to recover the local network structure of the networks given perfect Markov blanket information. Results are given is the form "mean $\pm$ standard deviation over the 9 data sets." The best performer for each type of structural error has been highlighted in bold. All runs of PC were done with a forced maximum size of the conditioning set of 6. The dagger (†) notes PC results from a single data set instead of 9 because of the long completion times. Represented graphically in Figure 9.

5.3 RFE-Based Approach

In this series of experiments, we tested our RFE-based approach on the Alarm network with sample sizes $n = 100$, 200, 300, 400 and 500. Table 3 lists the results and shows the number of errors as measured at different stages of the algorithm:

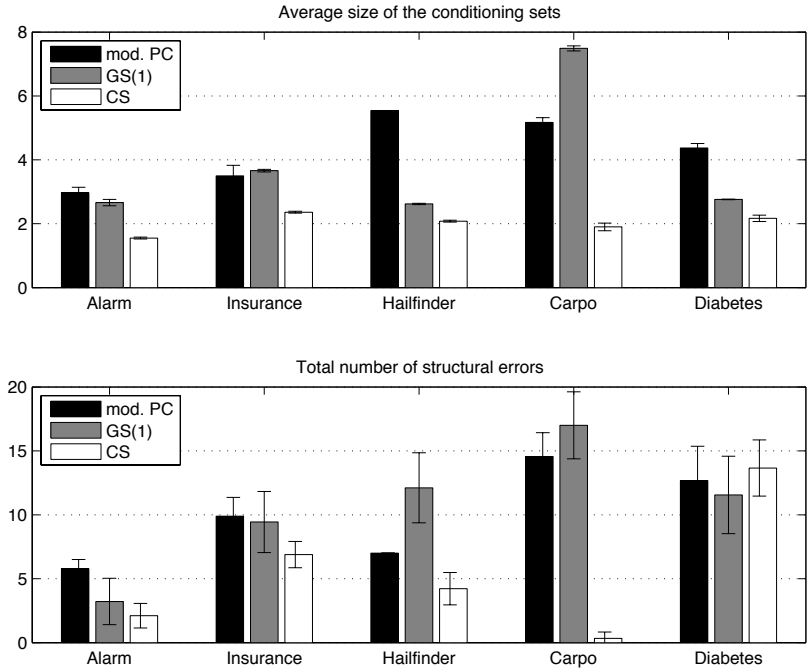

Figure 9: Average size of the conditioning sets and total number of errors for the three local structure discovery algorithms on various networks. Graphical representation corresponding to the results in Table 2.

1. Right after the Markov blanket identification, without adjustment. This compares the true Markov blanket of each variable with the identified Markov blanket as returned by Algorithm 5;

2. After building the moral graph. This notably excludes variables from Markov blankets if they do not satisfy the symmetry condition (2) due to the symmetry check performed at line 5 in the generic approach described in Algorithm 3;

3a. After removal of the spouse links using the Recursive Median (RM) algorithm (Margaritis, 2005) to check for conditional independence in the continuous domain;

3b. Alternatively, after removal of the spouse links using a test on Fisher's z-transform of partial correlation;

4a. After removal of the spouse links using RM *and* after maximal orientation. This is actually the result that can be compared to other full structure-learning algorithms;

4b. After removal of the spouse links using partial correlation tests *and* after maximal orientation;

5. Finally, we show how PC performs on the same instance for comparison.

Note that the RM test is a Bayesian distribution-free conditional-independence test. In this case, where we use multivariate Gaussian distributed data, we do not expect it to perform better

than the specialized z-test. We nevertheless include it in this series of experiments for two reasons. First, it allows the collider set search to be also distribution-free, in the sense that if "distribution-free feature selection" can be performed efficiently and consistently in the first phase, applying a subsequent collider set search does not make more assumptions on the distribution. Second, it allows to evaluate the cost of using a distribution-free algorithm on Gaussian data.

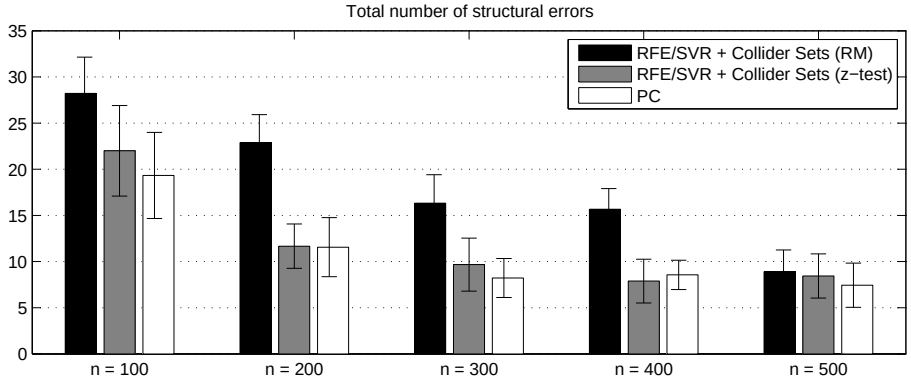

Figure 10: Total number of errors for the CPDAGs returned by the three global structure discovery algorithms on the Alarm network with various sample sizes. Graphical representation corresponding to the results in Table 3.

Detailed results are in Table 3 and the total number of structural errors is shown graphically in Figure 10. What we can read from the results is that, generally, the selected Markov blankets contain all variables from the true Markov blanket plus one or two additional variables. Starting at $n = 300$, on average, less than two variables were missed. Many spurious variable are selected, however, even for the larger data sets. This confirms the expectation the RFE approach also selects weakly relevant features: on average, the Markov blankets in the Alarm network have a size of 3.5, and on average 5.5 variables are selected per variable.

The symmetry check requiring Y to be part of $\mathbf{Mb}(X)$ and X to be part of $\mathbf{Mb}(Y)$ to add a link between X and Y fulfills its purpose, as even in the case $n = 200$ where on average about 73 variables enter wrong Markov blankets, only 4 extra links are added in the moral graph. As a side note, we thus argue that a global analysis can be beneficial to achieve better results on local tasks: we see here that determining via RFE the Markov blanket of a single variable is too inclusive, but that validating the selected variables globally, for instance with our Markov blanket symmetry check, allows to significantly reduce the number of false positives.

After the collider set search, the number of missing and extra arcs can both either increase or decrease. If the number of missing links increases, it is because the collider set search found d-separation too often while variables were actually dependent. If it decreases, it means that the missing arcs in the moral graph were spouse links, as their absence is not penalized in the PDAG any more. If the number of extra arcs increases, then the collider set search failed to identify spouse links; if it decreases, then the collider set search also removed through appropriate conditioning links that were not spouse links (which in turn possibly led to wrong orientations). Also, determining which part of the algorithm is responsible for a missed, extra, or reversed edge in a PDAG or CPDAG is not evident. As the feature-selection step is not alone responsible for the extra or missing links,

Stage	$n = 100$	$n = 200$	$n = 300$	$n = 400$	$n = 500$
1. Mb$(\cdot)$ ident.					
missing variables	17.33 ± 3.33	3.33 ± 1.48	0.78 ± 1.11	0.78 ± 1.48	0.33 ± 1.48
extra variables	80.66 ± 15.17	72.89 ± 9.25	74.78 ± 13.69	72.56 ± 9.99	73.33 ± 9.99
2. Moral graph					
missing arcs	23.44 ± 3.54	7.89 ± 3.48	4.33 ± 3.91	4.56 ± 3.47	4.33 ± 3.87
extra arcs	4.67 ± 2.24	4.11 ± 1.69	3.78 ± 1.20	3.11 ± 1.62	3.89 ± 2.20
TOTAL	28.11 ± 3.82	12.00 ± 2.55	8.11 ± 4.01	7.67 ± 4.06	8.22 ± 4.38
3a. PDAG/RM					
missing arcs	17.44 ± 2.35	10.00 ± 1.80	6.89 ± 2.67	6.33 ± 2.60	4.56 ± 1.51
extra arcs	4.78 ± 2.17	4.22 ± 1.56	3.33 ± 1.12	3.22 ± 1.48	3.44 ± 2.01
reversed arcs	1.22 ± 1.20	2.11 ± 1.27	3.11 ± 0.78	1.78 ± 0.97	0.67 ± 0.60
TOTAL	23.44 ± 1.67	16.33 ± 3.12	13.33 ± 2.65	11.33 ± 2.78	8.67 ± 2.37
3b. PDAG/z-t.					
missing arcs	11.89 ± 2.32	3.33 ± 1.32	2.67 ± 1.94	2.11 ± 1.17	2.56 ± 1.51
extra arcs	5.22 ± 2.22	4.00 ± 1.58	3.22 ± 1.20	2.78 ± 1.30	3.44 ± 2.01
reversed arcs	0.33 ± 0.50	0.56 ± 0.73	1.22 ± 0.67	0.78 ± 1.09	0.44 ± 1.13
TOTAL	17.44 ± 3.32	7.89 ± 2.15	7.11 ± 2.98	5.67 ± 2.12	6.44 ± 2.40
4a. CPDAG/RM					
missing arcs	17.44 ± 2.35	10.00 ± 1.80	6.89 ± 2.67	6.33 ± 2.60	4.56 ± 1.51
extra arcs	4.78 ± 2.17	4.22 ± 1.56	3.33 ± 1.12	3.22 ± 1.48	3.44 ± 2.01
reversed arcs	6.00 ± 3.87	8.67 ± 2.12	6.11 ± 2.32	6.11 ± 3.59	0.89 ± 0.60
TOTAL	28.22 ± 3.93	22.89 ± 3.02	16.33 ± 3.08	15.67 ± 2.24	8.89 ± 2.37
4b. CPDAG/z-t.					
missing arcs	11.89 ± 2.32	3.33 ± 1.32	2.67 ± 1.94	2.11 ± 1.17	2.56 ± 1.51
extra arcs	5.22 ± 2.22	4.00 ± 1.58	3.22 ± 1.20	2.78 ± 1.30	3.44 ± 2.01
reversed arcs	4.89 ± 3.33	4.33 ± 1.73	3.78 ± 1.09	3.00 ± 1.80	2.44 ± 1.13
TOTAL	22.00 ± 4.90	$\mathbf{11.67} \pm 2.40$	9.67 ± 2.87	$\mathbf{7.89} \pm 2.37$	8.44 ± 2.40
5. CPDAG/PC					
missing arcs	12.11 ± 2.52	7.44 ± 1.42	4.22 ± 0.97	5.67 ± 1.12	4.78 ± 0.83
extra arcs	4.56 ± 2.19	2.67 ± 1.87	2.78 ± 1.48	2.11 ± 0.93	2.00 ± 1.66
reversed arcs	2.67 ± 1.80	1.44 ± 1.51	1.22 ± 1.09	0.78 ± 1.09	0.67 ± 0.87
TOTAL	$\mathbf{19.33} \pm 4.66$	$\mathbf{11.56} \pm 3.2$	$\mathbf{8.22} \pm 2.11$	8.56 ± 1.59	$\mathbf{7.44} \pm 2.40$

Table 3: Structural errors at various stages of the RFE-based approach, showing the missing, extra and reversed arcs with respect to the original graph. For Step 1, identification of the Markov blanket, the figures are averages over the 37 variables; that is, the count of the extra or missing variables per Markov blanket, and thus not directly comparable to the other steps. The sums of the errors for the CPDAGs are represented in Figure 10.

the collider set search is not responsible for all orientation mistakes. In the collider set search, if a wrong spouse link is removed, it is because a wrong V-structure has been identified, so that the absence of an arc will be linked to the wrong orientation of the falsely recognized V-structure. It is also possible to construct cases where missing a variable in the feature-selection step will lead not only to a missing arc, but also to the detection of a spurious V-structure, even if all subsequent tests are perfect.

For the PDAGs obtained using z-tests, the number of missing arcs always decreases with respect to the moral graph, and so does the number of extra links for $n \geq 200$. We find that the more general RM test seems to return independence too often, as for $n \geq 200$ more links are missing in the PDAG than in the moral graph. (On highly nonlinear data, we would, however, expect RM to perform better than a z-test, which assumes Gaussianity.)

The CPDAGs do not have a number of adjacency errors different from their PDAGs; this step can only add directionality errors. We have nevertheless copied the results in order to improve the readability and to make the comparison with PC easier. Although the RFE approach can outperform PC in adjacency errors, PC still consistently makes fewer directionality errors. We remark, however, that the overall performance of RFE/SVR with z-tests is very comparable to that of PC, as also shown in Figure 10, which empirically justifies the intuition behind this approach.

5.4 TC and TC_{bw} vs. Competitors

For this series of experiments, we performed more systematic testing of TC, TC_{bw}, PC, the full GS and the Bach-Jordan method on data sets sampled from Alarm, Insurance, Hailfinder, Carpo, and Diabetes, varying the sample size. The Bach-Jordan method consists of a scoring function based on Mercer kernels coupled with a greedy search in the space of DAGs and was designed to learn Bayesian networks. It does not guarantee that the formal semantics of a causal graph are respected in the large-sample limit, but has been included in the experiments for the sake of comparison. Other possible competitors like SCA (Friedman et al., 2000) or AlgorithmMB (Peña et al., 2005) were inapplicable because generalizing them to handle continuous variables require techniques that are too computationally expensive, notably because of score-based subroutines that are hard to generalize.

The structural errors, like before, are missing, extra, and reversed arcs in the returned CPDAG with respect to the generating graph. For the Bach-Jordan method, similarly to what was done in Fu (2005), we converted the returned DAG to its essential graph first before checking for structural errors to avoid penalizing statistically equivalent structures. For all experiments, we also compare the run times and the number of tests performed by TC, TC_{bw}, GS, and PC.

Specific to the Bach-Jordan method is the issue of choosing the appropriate kernel parameters; that is, in our case, the σ width in the Gaussian kernel. Bach and Jordan (2003) claim that the algorithm is in general robust to the choice of σ. We have found, however, that for varying sample sizes, the number of structural errors is quite sensitive to σ. As the authors do not propose a heuristics to set it, we systematically tested the algorithm with $\sigma = 2$, 1, 0.5, and 0.3 for each run, and chose the outcome with the smallest sum of structural errors. In general, smaller data sets preferred $\sigma = 2$, while the larger ones preferred a smaller σ. The change of σ is not directly visible in the following plots of the errors, but it often leads to "zigzags" in the Bach-Jordan curves. This is due to the fact that we only tested a fixed number of values for σ and did not perform a full optimization of this parameter for each run. The results shown are thus not the best results obtainable with this method.

5.4.1 ALARM

The figures on p. 1330 show the structural errors, run times and number of statistical tests against the number of samples for Alarm. For each sample size, 5 data sets were drawn from the model; the error bars picture the standard deviation over these 5 runs.

The numbers of extra and missing links seem to clearly decrease on average for all algorithms with an increasing number of samples, except for Bach-Jordan, which sometimes has the tendency to add more links when more data points are available. Note that Bach-Jordan's σ changes between the last two runs, explaining the abrupt change in the extra arcs. The number of reversed arcs seems to less satisfactory, in particular for TC. The explanation is that TC misses many arcs with low sample sizes, and thus does not actually get the opportunity to make many directionality errors for these cases. TC_{bw} exhibits a related behavior, although much less stronger. We also see that Bach-Jordan makes the most directionality errors (this is actually valid for all networks). GS reaches repeatedly a zero extra arc score for $n > 1000$, although it misses some more than the others.

Starting at about 200 samples, TC equals or outperforms PC, GS and Bach-Jordan. TC_{bw} beats both TC and PC, and the converging curves of TC and TC_{bw} show that the stepwise regression becomes unnecessary with about 400 samples. On average, TC was about 20 times faster than the implementation of PC we used, although the factor tended to decrease with larger sample sizes. TC_{bw} was naturally slower than TC, although only marginally compared to the speed difference with PC.

Overall, the constraint-based methods seem to perform approximately equally well for $n > 400$, and TC_{bw} and GS perform slightly better than PC for low sample sizes. Note that for low sample sizes, TC is always outperformed by TC_{bw}, PC, and GS, but is often the one to perform best when the sample size gets larger. The graphs in Figure 12 show that TC and TC_{bw} are fastest, although GS performed fewer tests that TC_{bw}.

5.4.2 INSURANCE

For this network, we find similar behaviors to Alarm, shown in the graphs on p. 1331. The most striking difference is the clear tendency of Bach-Jordan to add more arcs when more data is available for this more densely connected network. Between the 5th and 6th sample sizes, there is again a change of σ. Comparing the curves of the missing and extra arcs, we see that this changes the tradeoff between false negatives and false positives.

In this case, too, TC_{bw} outperforms TC with low sample sizes (because it misses fewer arcs) but is outperformed with bigger data sets (because it adds too many). Both PC and GS, while being slightly better than TC_{bw} for $n < 100$, are outperformed starting at about $n = 500$. Note the overall good performance of GS in terms of arc orientation errors. The corresponding curve also decreases more smoothly with larger data sets. The Bach-Jordan method is unexpectedly fast on this data set, although poorly accurate. The pattern of the number of statistical tests is very similar to that of Alarm.

5.4.3 HAILFINDER

This network poses a problem to PC: we divided its run time and the number of tests by 10 in the graphs of Figure 16 on p. 1332. Because of its long run times, PC was run only once for each point in the plots, so that the error bars are missing. PC runs into trouble because of the node cluster around variable 27 in the network (see Figure 6): it tries to separate it from the other nodes by doing

subset searches on its large number of neighbors. In order to speed it up, we set the maximum node fan-in parameter to 6, so that PC would not attempt to conduct conditional-independence tests with conditioning sets larger than 6 (we see in Figure 16 how this imposes an upper bound on the run times of PC). TC and TC_{bw} do not run into this problem, because this cluster is correctly left alone after the feature-selection step, done in $O(d^3)$ operations. Note that TC, TC_{bw}, and GS *would* also spend a long time on this cluster if all neighbors of variable 27 were its parents, because they would contain a lot of extra spouse links to be checked with an exponential number of combinations. But this example shows that a local lack of sparseness is fatal to the efficiency of PC, whereas other algorithms can still deal with it if the density of the connections is caused by children rather than parents.

This network shows more clearly the missing arc problem that TC has with low sample size, and the benefits of using TC_{bw} rather than TC here, at least for $n < 2000$. On this network, GS performs overall well. It is beaten by TC only for $n > 2000$, but performs better than all others for $n < 200$. Bach-Jordan still exhibits the same tendency to add more arcs when more data is available. For this data set, σ changes twice, between the 4th and 5th, and between the 5th and 6th data set sizes. The 5th sample size seems to have generated an unfavorable data set for PC, as the number of extra arcs is particularly high.

Examining the run times designates TC as the fastest. This is important especially with larger sample sizes, as TC is often both the fastest and most accurate algorithm.

5.4.4 CARPO

The results for this network are shown on p. 1333. The structural particularity of this network is multiple cases of a single variable having many children. PC performs overall badly on this network. For $n < 200$, GS is the clear winner: all other algorithms make many more errors. The plain TC especially misses many arcs. For $n > 500$, however, both TC and TC_{bw} slightly but consistently outperform GS. At $n = 800$, TC beats TC_{bw}. Bach-Jordan, although fast on this instance, adds again too many extra links, and makes numerous directionality errors.

5.4.5 DIABETES

This is our largest and final test network. The error patterns are most similar to those of the Insurance network, with the exception of Bach-Jordan, which performs more poorly here. We can detect two changes of σ: between the 3rd and 4th, and between the 5th and 6th sample sizes.

Starting at $n > 1000$, all constraint-based methods seem to yield similar overall accuracy. GS is better in terms of directionality errors; TC and TC_{bw} are better in terms of missed links. For $n > 4000$, TC and TC_{bw} have the same accuracy and slightly beat GS and PC, while they are beaten significantly for $n < 800$. We note that the extra links added by GS seem to allow it to obtain a better directionality accuracy than in our first series of experiments, where it was given the exact moral graph as input.

5.4.6 DISCUSSION

Both TC and TC_{bw} slightly but consistently beat the other competitors when the sample size exceeds one or two thousand, depending on the network. They are usually weaker with low sample sizes because of missed arcs. GS beats TC with small data sets, because of the way that PC goes through conditioning sets for the statistical tests (Tsamardinos et al., 2006, discuss in detail this particular

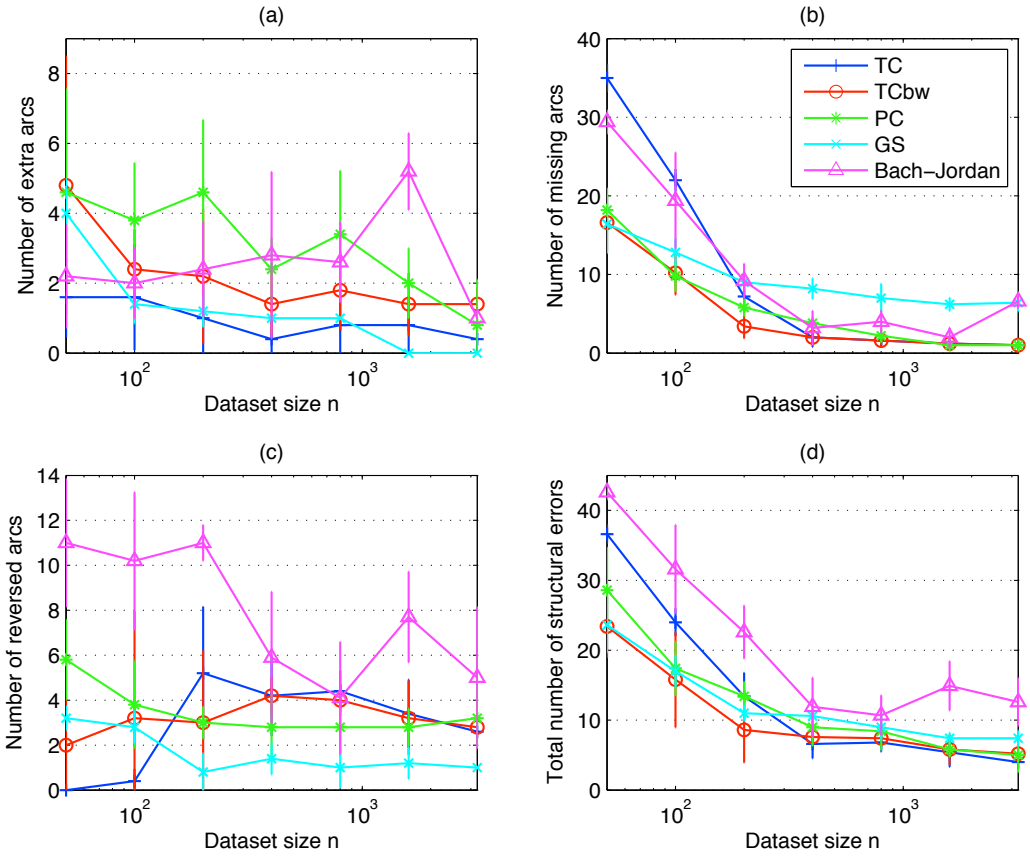

Figure 11: Differentiated errors on Alarm as a function of the sample size n: (a) extra arcs; (b) missing arcs; (c) reversed arcs; (d) total sum.

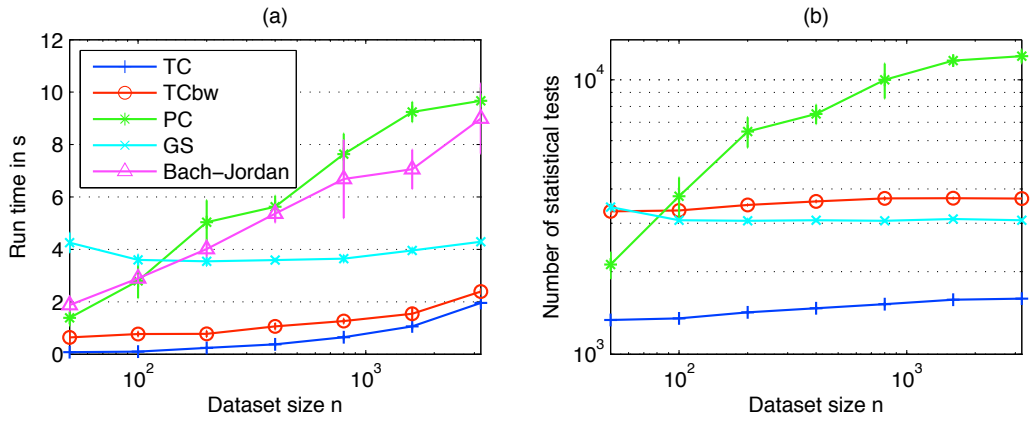

Figure 12: Alarm: (a) run times and (b) number of statistical tests as a function of the sample size n.

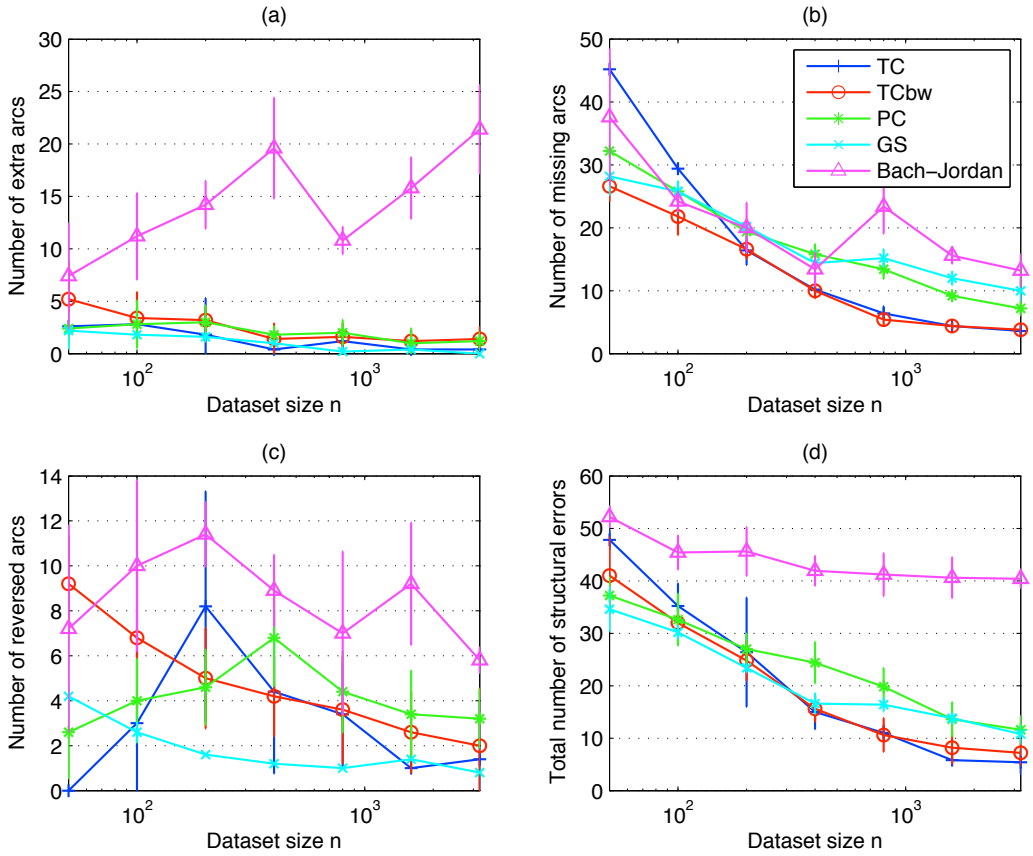

Figure 13: Differentiated errors on Insurance as a function of the sample size n: (a) extra arcs; (b) missing arcs; (c) reversed arcs; (d) total sum.

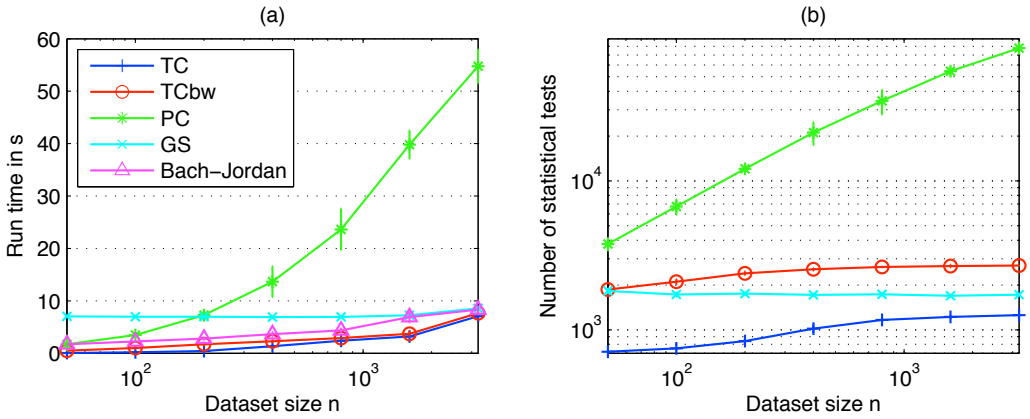

Figure 14: Insurance: (a) run times and (b) number of statistical tests as a function of the sample size n.

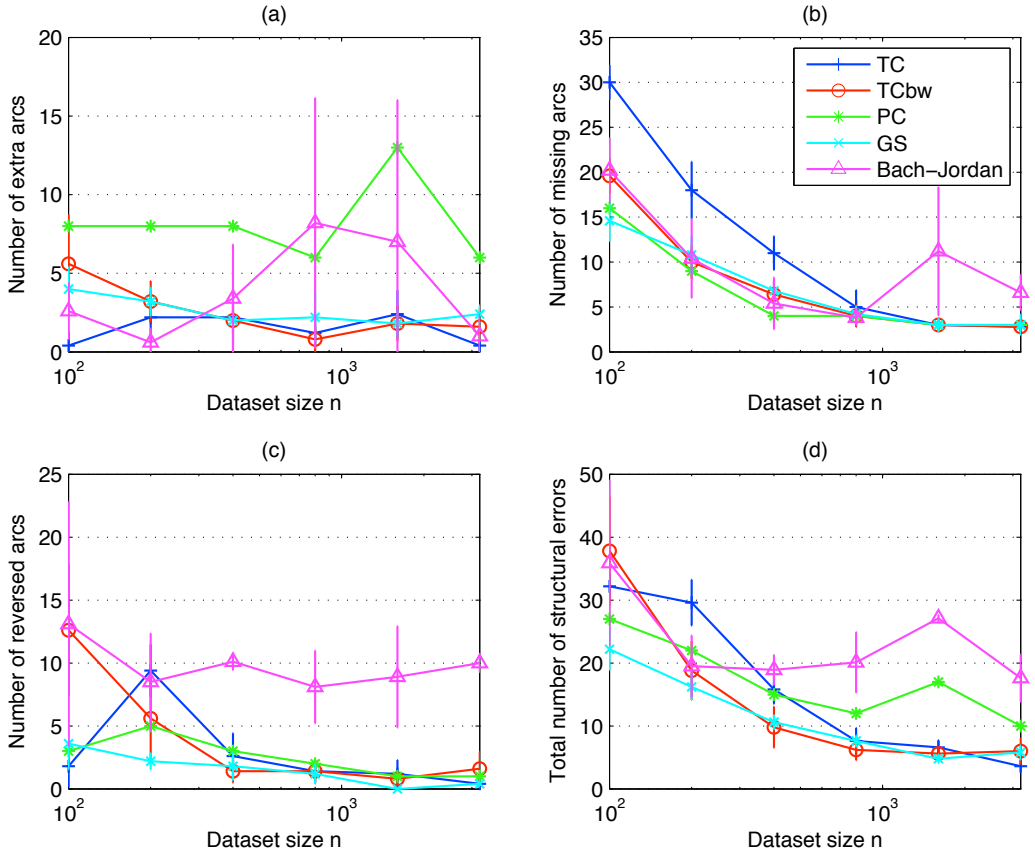

Figure 15: Differentiated errors on Hailfinder as a function of the sample size n: (a) extra arcs; (b) missing arcs; (c) reversed arcs; (d) total sum.

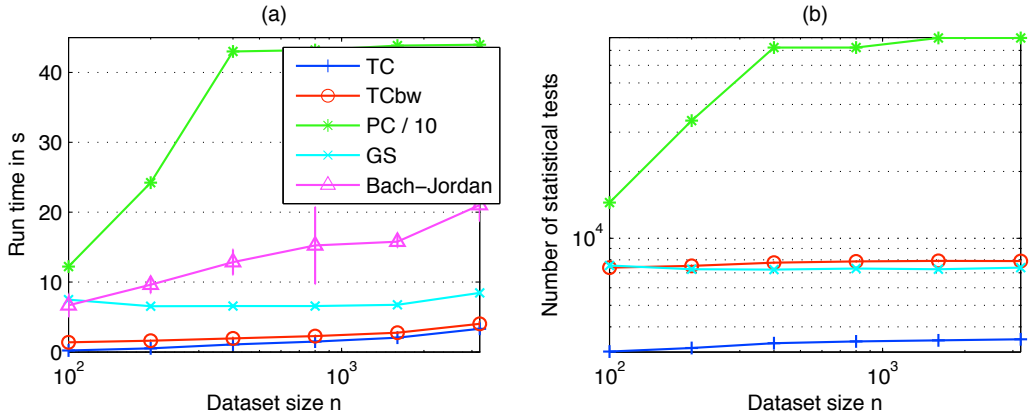

Figure 16: Hailfinder: (a) run times and (b) number of statistical tests as a function of the sample size n.

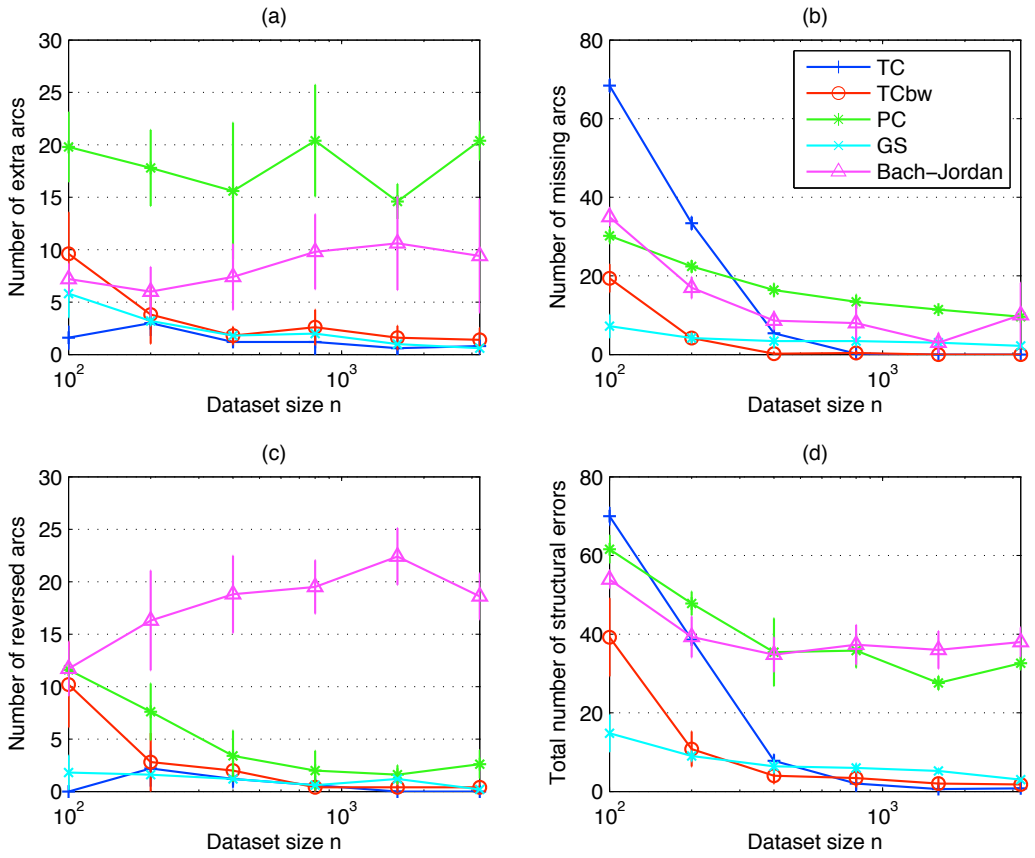

Figure 17: Differentiated errors on Carpo as a function of the sample size n: (a) extra arcs; (b) missing arcs; (c) reversed arcs; (d) total sum.

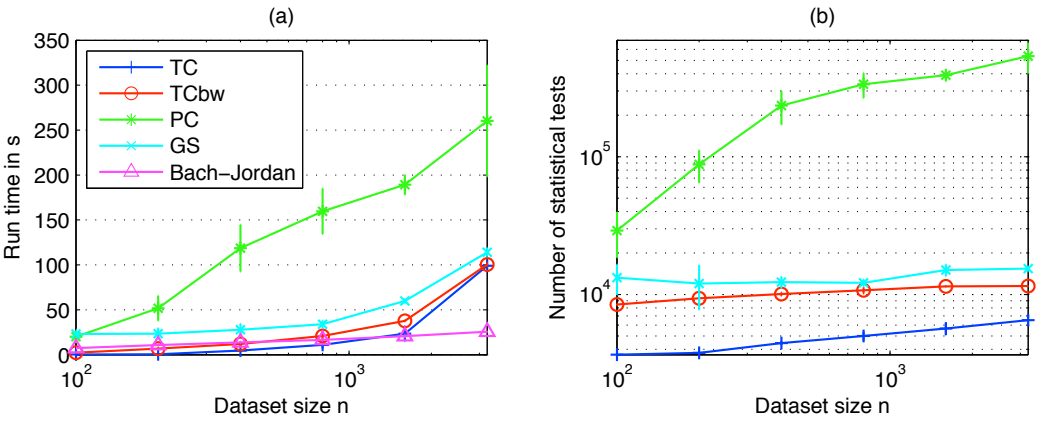

Figure 18: Carpo: (a) run times and (b) number of statistical tests as a function of the sample size n.

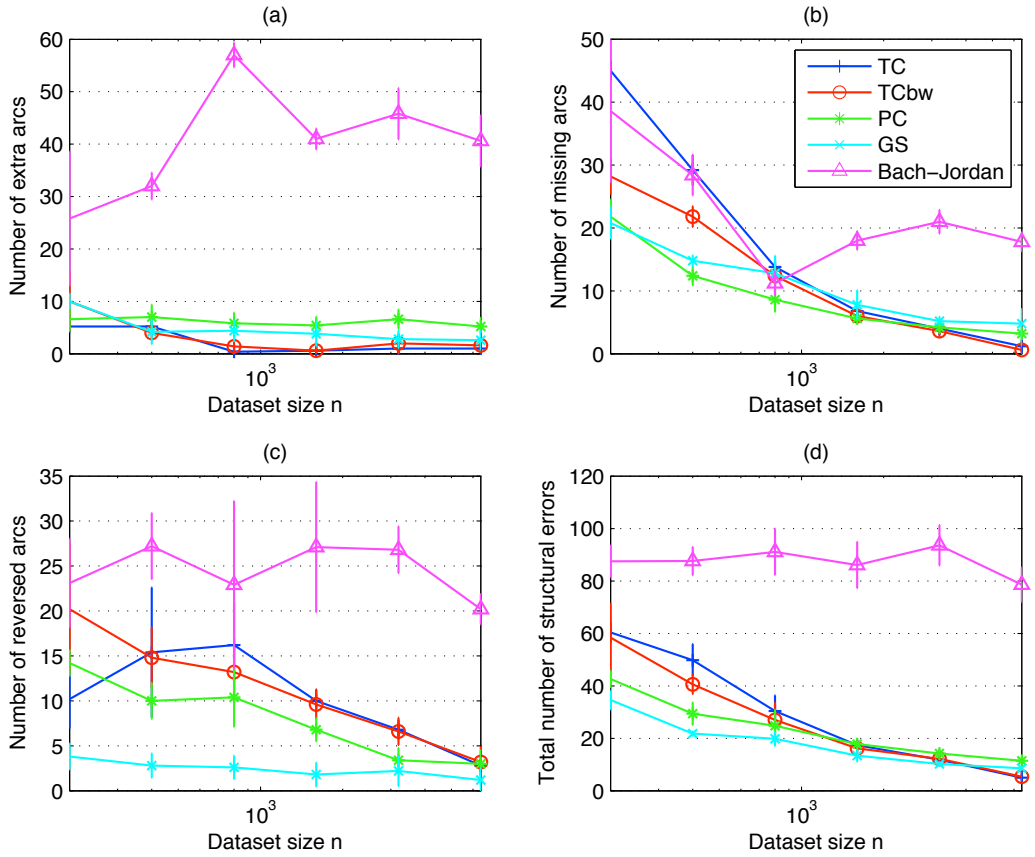

Figure 19: Differentiated errors on Diabetes as a function of the sample size n: (a) extra arcs; (b) missing arcs; (c) reversed arcs; (d) total sum.

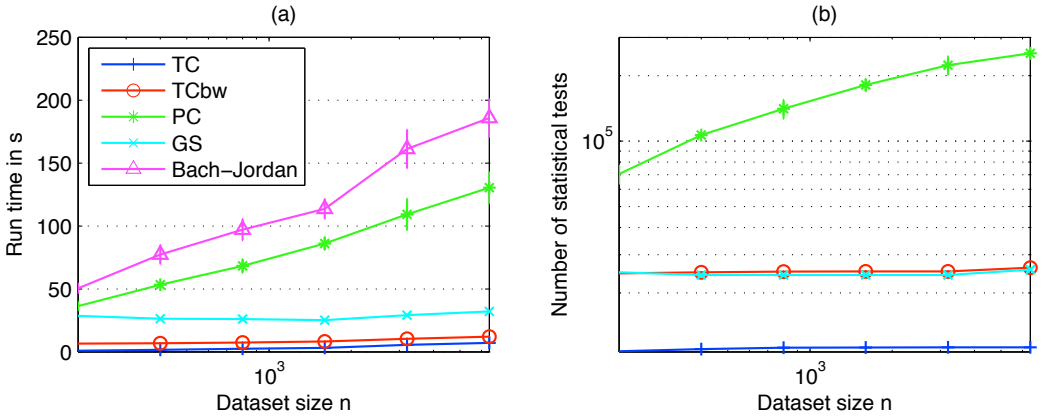

Figure 20: Diabetes: (a) run times and (b) number of statistical tests as a function of the sample size n.

issue in the case of tests with discrete variables). The score-based Bach-Jordan method was found difficult to tune with the parameter σ. For this multivariate Gaussian case, its performance is usually worse than the other tested algorithms. This also reflects the fact PC, GS, TC and TC_{bw} with z-tests are all "tuned" for multivariate Gaussian data. The additional errors made by Bach-Jordan reflect the price of being more generic.

In terms of run time, PC is slowed down by nodes with a high degree, whereas TC or GS handle them without the exponential time complexity growth if they are not part of triangles, as in Hailfinder. In general, TC and TC_{bw} resolve all conditional-independence relations (up to married parents) in the feature-selection step in $O(d^3)$ and $O(d^4)$, respectively, whereas all PC can do in $O(d^{2+\alpha})$ is resolve conditional-independence relations with conditioning sets of cardinality α. It is then reasonable to expect algorithms like GS, TC and TC_{bw} to scale better than PC on sparse networks where nodes have a small number of parents. The exponential growth in PC can be seen in case the nodes have a high degree, be it parents or children; in TC and GS, it is due to large fully-connected triangle structures and to spouse links coming from the Markov blanket-construction step. And whereas these large structures imply a high degree, the converse is not true (for instance in the Hailfinder network). So, PC will exhibit an exponential behavior on all problem instances where TC and GS also exhibits this behavior, but the converse is not true.

It is interesting to investigate what kind of high-degree structure is more likely to appear. If it is a node with many children (as node 27 in Hailfinder), which we call an *explosion pattern*, TC can handle it efficiently. If it is a node with many parents, an *implosion pattern*, then none of these algorithms can recognize it in polynomial time. Explosion patterns correspond to a single cause that has many effects; implosion patterns correspond to many causes leading to the same effect. It remains open for discussion to know which one is more likely to occur with real-life data sets.

GS could not be beaten on small sample sizes. It is yet an unsolved challenge for TC and TC_{bw} to handle problems where the number of variables exceeds the number of samples, as in gene expression networks, thus leading to an attempt at inverting a matrix that does not have full rank. Regularizing the covariance matrix might help make TC_{bw} more robust in this case. Computationally, TC_{bw} does add a degree of complexity with respect to TC, and the number of tests that TC_{bw} performs is usually comparable to GS.

TC_{bw} helps solving problems with TC and small data sets, but still cannot operate below the $n = d$ threshold. The exact sample size where TC_{bw} stops performing better than TC does not appear to be a simple function of the n or d but depends on the structure of the network. It would be useful to investigate when the feature-selection addition of TC_{bw} becomes irrelevant. And as GS is more accurate with small sample sizes, finding a similarly testable condition predicting the threshold where TC is more accurate than GS would allow to merge the approaches into a single algorithm that knows which Markov blanket approach to use in order to achieve better results.

6. Conclusion

Causal discovery and feature selection are closely related: optimal feature selection discovers Markov blanket as sets of strongly relevant features, and causal discovery discovers Markov blankets as direct causes, direct effects, and common causes of direct effects. By performing perfect feature selection on each variable, we get the undirected moral graph as an approximation of the causal graph. An extra step, the collider set identification, is needed in order to transform the Markov blankets into parents, children, and spouses. This step is exponential in the worst case,

but is actually efficient provided the graph is sparse enough—a common assumption of many algorithms. We proposed an algorithm to do this task and compared it favorably to the similar steps of the Grow-Shrink algorithm.

Determining the Markov blanket with existing backward feature elimination like RFE eliminates the irrelevant variables in the large sample limit, but remains too inclusive. Global corrections have to be made, such as for instance insuring that a variable in the selected Markov blanket of a target also includes this target in its own selected Markov blanket. We conducted experiments that confirmed that this adjustment discards most false positives, and thus provided a hint that the approach is consistent in the large-sample limit. The main challenge is to perform feature selection for all variables in an efficient way. This task is tractable with the multivariate Gaussian assumption. We presented the TC and the TC_{bw} algorithms, which fit into the described framework, and compared them to PC, GS, and a Bayesian structure-learning method. For small sample sizes, GS usually makes fewer structural errors, and TC/TC_{bw} are better for larger samples sizes.

We are convinced of the superiority of the Markov blanket approaches as described in this paper. We invoke as support for this claim the high run times of PC, and the good low and high sample size accuracy of GS and TC/TC_{bw}, respectively. Not only are Markov blanket techniques much more scalable, they can be more accurate; they are also more easily modifiable to construct only parts the network deemed relevant by some criterion.

The biggest challenges we face now with causal structure learning include robust and consistent distribution-free structure learning with continuous and potentially highly nonlinear data. In the future, we intend to make use of this framework to develop such techniques and thus try to get rid of the Gaussianity assumption, often impractical with real-life data sets.

Acknowledgments

We would like to thank Dimitris Margaritis, who kindly provided us with his C implementation of the Recursive Median conditional-independence test; and Francis Bach and Lawrence Fu, who provided us with a Matlab/C implementation of the Bach-Jordan algorithm. We also thank the anonymous reviewers for their helpful comments and pointers, which led to a significantly enhanced version of this paper.

Appendix A.

For all proofs, we assume the given data set D is faithful.

Lemma 12 *In a DAG $\mathcal{G}$, any (undirected) path π of length $\ell(\pi) > 2$ can be blocked by conditioning on any two consecutive nodes in π.*

Proof It follows from Definition 5 that a path π is blocked when either at least one collider (or one of its descendants) is not in the conditioning set $\mathbf{S}$, or when at least one non-collider is in $\mathbf{S}$. It therefore suffices to show that conditioning on two consecutive nodes always includes a non-collider. This is the case because two consecutive colliders would require bidirected arrows, which is a structural impossibility with simple DAGs. ■

Lemma 13 *In a DAG $\mathcal{G}$, two nodes X,Y are* d*-connected given all other nodes $\mathbf{S} = \mathbf{V} \setminus \{X,Y\}$ if and only if any of the following conditions holds:*

(i) There is an arc from X to Y or from Y to X (i.e., $X \to Y$ or $X \leftarrow Y$);
(ii) X and Y have a common child Z (i.e., $X \to Z \leftarrow Y$).

Proof We prove this by first proving an implication and then its converse.

($\Longleftarrow$) If (i) holds, then X and Y cannot be d-separated by any set. If (ii) holds, then Z is included in the conditioning set and d-connects X and Y by Definition 5.

($\Longrightarrow$) X and Y are d-connected given a certain conditioning set when at least one path remains open. Using the conditioning set $\mathbf{S}$, paths of length > 2 are blocked by Lemma 12 since $\mathbf{S}$ contains all nodes on those paths. Paths of length 2 contain a mediating variable Z between X and Y; by Definition 5, $\mathbf{S}$ blocks them unless Z is a common child of X and Y. Paths of length 1 cannot be blocked by any conditioning set. So the two possible cases where X and Y will be d-connected are (i) or (ii). ■

Corollary 14 *Two variables X,Y are dependent given all other variables $\mathbf{S} = \mathbf{V} \setminus \{X,Y\}$ if and only if any of the following conditions holds:*

(i) X causes Y or Y causes X;
(ii) X and Y have a common effect Z.

Proof It follows directly from Lemma 13 due to the faithful structure, which ensures that there exists a DAG where conditional independence and d-separation map one-to-one. Lemma 13 can then be reread in terms of conditional independence and causation instead of d-separation and arcs. ■

Property 7 (Total conditioning) *In the context of a faithful causal graph $\mathcal{G}$, we have:*

$$\forall X,Y \in \mathbf{V}: \ \big(X \in \mathbf{Mb}(Y) \iff (X \not\perp\!\!\!\perp Y \mid \mathbf{V} \setminus \{X,Y\})\big).$$

Proof This is a direct consequence of Corollary 14, where points *(i)* and *(ii)* lead to the definition of the Markov blanket of Y as *(i)* all its causes and effects, and *(ii)* the other direct causes of its effects. This is equivalent to $\mathbf{Mb}(Y)$ in $\mathcal{G}$. ■

Lemma 15 *When it exists, the subset $\mathbf{Z}$ that has the Collider Set property for the pair (X,Y) is the set of all direct common effects of X and Y.*

Proof We prove this using $\mathbf{Z}$ and a corresponding $\mathbf{S}_{XY}$ that fulfills (7).

($\Longrightarrow$) ($Z_i \in \mathbf{Z} \implies X \to Z_i \leftarrow Y$.) By (7) and (8), we know that each Z_i opens a dependence path between X and Y (which are independent given $\mathbf{S}_{XY}$) by conditioning on $\mathbf{S}_{XY} \cup \{Z_i\}$. By Definition 5, conditioning on Z_i opens a path if Z_i is either a colliding node or one of its descendants. As, by

definition, $\mathbf{Z} \subseteq \mathbf{Tri}(X - Y)$, we are in the first case. We conclude that Z_i is a direct effect of both X and Y.

($\Longleftarrow$) ($X \twoheadrightarrow Z_i \twoheadleftarrow Y \implies Z_i \in \mathbf{Z}$.) Note that (7) and (8) together are implied in presence of a V-structure $X \twoheadrightarrow Z_i \twoheadleftarrow Y$. Thus, a direct effect is compatible with the conditions. The fact that $\mathbf{Z}$ captures all direct effects follows from the maximization of its cardinality. ■

Lemma 9 *In the context of a faithful causal graph, the set* $\mathbf{Z}$ *that has the Collider Set property for a given pair* (X,Y) *exists if and only if* X *is neither a direct cause nor a direct effect of* Y*, and is unique when it exists.*

Proof The fact that $\mathbf{Z}$ exists if and only if X is neither a direct cause nor a direct effect of Y is a direct consequence of (7), which states that X and Y can be made conditionally independent. This is in contradiction with direct causation.

We now show unicity, using interchangeably the criteria of d-separation and conditional independence, as allowed by the Faithfulness assumption. Suppose that, for a pair (X,Y), two sets $\mathbf{Z}$, $\mathbf{W}$ have been found that fulfill the Collider Set property, with the corresponding d-separating sets $\mathbf{S}_{XY}^{\mathbf{Z}} \subseteq \mathbf{V} \setminus \{X,Y\} \setminus \mathbf{Z}$ and $\mathbf{S}_{XY}^{\mathbf{W}} \subseteq \mathbf{V} \setminus \{X,Y\} \setminus \mathbf{W}$ fulfilling (7). Let $\mathbf{Z}^\star = \mathbf{Z} \setminus \mathbf{W}$. Due to symmetry, proving that $\mathbf{Z}^\star$ is empty proves that $\mathbf{Z} = \mathbf{W}$.

Suppose that $\mathbf{Z}^\star \neq \emptyset$; that is, $\exists Z \in \mathbf{Z}^\star$. Then, by definition, we have that $\left(X \perp\!\!\!\perp Y \;\middle|\; \mathbf{S}_{XY}^{\mathbf{Z}}\right)$ and $\left(X \not\perp\!\!\!\perp Y \;\middle|\; \mathbf{S}_{XY}^{\mathbf{Z}} \cup \{Z\}\right)$. We now have two cases: either (i) $Z \notin \mathbf{S}_{XY}^{\mathbf{W}}$, or (ii) $Z \in \mathbf{S}_{XY}^{\mathbf{W}}$. In the former case (i), consider the set $\mathbf{W}' = \mathbf{W} \cup \{Z\}$. Then $\mathbf{W}'$ also fulfills the Collider Set property with the same d-separating set $\mathbf{S}_{XY}^{\mathbf{W}}$: the only additional condition is $\left(X \not\perp\!\!\!\perp Y \;\middle|\; \mathbf{S}_{XY}^{\mathbf{W}} \cup \{Z\}\right)$. This holds because, as shown by Lemma 15, Z is a direct child of X and Y, and conditioning on it opens a path, no matter what the conditioning set is. But all this is in contradiction with the definition stating that any set fulfilling this property must be the largest set to do so, because the cardinality of $\mathbf{W}'$ is greater than that of $\mathbf{W}$.

In the latter case (ii), the d-separating set $\mathbf{S}_{XY}^{\mathbf{W}}$ contains Z. But this is impossible due to the same reason that Z is a direct child of both X and Y and that thus any set containing Z cannot d-separate X and Y. We therefore conclude $\mathbf{Z}^\star = \emptyset$ and $\mathbf{Z} = \mathbf{W}$, which leads to the uniqueness of the set fulfilling the Collider Set property. ■

Theorem 10 *In the large sample limit, for faithful, causally sufficient data sets, the procedure* RESOLVEMARKOVBLANKETS_COLLIDERSETS *correctly identifies all V-structures and all spouse links, assuming consistent statistical tests.*

Proof First, we note that in a moral graph, a node X is connected to its parents, children, and spouses. Thus, all spouse links to be removed are in the moral graph, and, by the definition of spouse, each spouse link between X and Y corresponds to at least one unshielded collider for the pair (X,Y). Additionally, by the definition of unshielded collider, X and Y are nonadjacent, so that for each spouse link $X - Y$ there is a set $\mathbf{S}_{XY}$ such that $(X \perp\!\!\!\perp Y \mid \mathbf{S}_{XY})$ by the contraposition of (4). So, when such a set $\mathbf{S}_{XY}$ is found, the link $X - Y$ is removed, and for each Z such that $X - Z - Y$ and $Z \notin \mathbf{S}_{XY}$, we orient the triplet as $X \rightarrow Z \leftarrow Y$ for the exact same reason that allows IC (or PC) to do the same in Step 2 of the algorithm (Pearl, 2000). The proof boils down to proving that the proposed search procedure always identify a d-separating set $\mathbf{S}_{XY}$ when there is one.

If some S_{XY} exists, then the link between X and Y is a spouse link by definition of a moral graph, which implies that X and Y have a nonempty set of common effects $\mathbf{Z}$. Each $Z \in \mathbf{Z}$ is linked to both X in Y and is thus in $\mathbf{Tri}(X-Y)$ by definition. Let us assume we can d-separate X and Y by some set: then, by the definition of d-separation, only conditioning on a common effect or a descendant of a common effect can create a dependency. In Algorithm 2, all possible colliders (line 7) and descendants of currently conjectured colliders (line 13) undergo a subset search, such that there will always be one iteration where all colliders and their descendants will be left out of the conditioning set. It is then enough to show that all d-connecting paths between X and Y that are not due to conditioning on a collider or collider's descendant go through the base conditioning set as determined at line 6.

To prove this, we note that the subset search at line 7 will always go through an iteration where it blocks all such d-connecting paths of length 2, that is, patterns of the type $X \to W \to Y$ and $X \leftarrow W \to Y$. As a direct consequence of the fact that we are working on the moral graph, all longer dependency paths go both through a node W in the set of immediate neighbors $\mathbf{Bd}(X)$ of X, and through a node in $\mathbf{Bd}(Y)$. Let us look at $\mathbf{Bd}(X)$. We have two cases: either (i) $W \in \mathbf{Tri}(X-Y)$ and will eventually be blocked by the subset search at line 7, or (ii) $W \in \mathbf{Bd}(X) \setminus \mathbf{Tri}(X-Y)$ (and thus $W \in \mathbf{Bd}(X) \setminus \mathbf{Tri}(X-Y) \setminus \{Y\}$ because $W \neq Y$). This set is exactly the set selected as base conditioning set at line 6, blocking all such paths, up to some symmetry with Y. The fact that we may choose the smaller of the two possible base conditioning sets is due to symmetry reasons. ■

Theorem 16 *If the variables are jointly distributed according to a multivariate Gaussian, TC returns the maximally oriented PDAG of the Markov equivalence class of the DAG representing the causal structure of the data-generating process in the large sample limit, assuming statistically consistent tests.*

Proof An edge is added between X and Y in the feature selection if we find that $\rho_{XY \cdot \mathbf{V} \setminus \{X,Y\}} \neq 0$. We conclude $(X \not\perp\!\!\!\perp Y \mid \mathbf{V} \setminus \{X,Y\})$ owing to the multivariate Gaussian distribution. Corollary 14 says that this implies that X causes Y or Y causes X, or that they share a common child. Therefore, each V-structure is turned into a triangle by the end of the feature-selection step. The collider set search then examines each link $X-Y$ part of a triangle, and by Lemma 15, we know that if the search for a set $\mathbf{Z}$ that has the Collider Set property succeeds, there must be no link between X and Y. We know by the same lemma that this set includes all colliders for the pair (X,Y), so that all V-structures are correctly identified. Step 3 is the same as in the IC or PC algorithms; see Pearl and Verma (1991) and Spirtes et al. (2001). ■

References

B. Abramson, J. Brown, A. Murphy, and R. L. Winkler. Hailfinder: A Bayesian system for forecasting severe weather. *International Journal of Forecasting*, 12:57–71, 1996.

C. F. Aliferis, I. Tsamardinos, and A. Statnikov. HITON, a novel Markov blanket algorithm for optimal variable selection. In *Proceedings of the 2003 American Medical Informatics Association (AMIA) Annual Symposium*, pages 21–25, 2003.

S. Andreassen, R. Hovorka, J. Benn, Kristian G. Olesen, and E. R. Carson. A model-based approach to insulin adjustment. In *Proc. of the Third Conf. on AI in Medicine*, pages 239–248. Springer-Verlag, 1991.

K. Baba, R. Shibata, and M. Sibuya. Partial correlation and conditional correlation as measures of conditional independence. *Australian & New Zealand Journal of Statistics*, 46(4), 2004.

F. R. Bach and M. I. Jordan. Learning graphical models with Mercer kernels. In *Advances in Neural Information Processing Systems 15*, 2003.

I. Beinlich, H. J. Suermondt, R. M. Chavez, and G. F. Cooper. The ALARM monitoring system: A case study with two probabilistic inference techniques for belief networks. In *Proc. of the Second European Conf. on AI in Medicine*, pages 247–256, 1989.

J. Binder, D. Koller, S. Russell, and K. Kanazawa. Adaptive probabilistic networks with hidden variables. *Machine Learning*, 29, 1997.

D. M. Chickering. Optimal structure identification with greedy search. *The Journal of Machine Learning Research*, 3:507–554, 2002.

G. Elidan. Bayes net repository. Website, 2001. URL `http://compbio.cs.huji.ac.il/Repository/`.

N. Friedman, M. Linial, I. Nachman, and D. Pe'er. Using Bayesian networks to analyze expression data. In *RECOMB*, pages 127–135, 2000.

L. D. Fu. A comparison of state-of-the-art algorithms for learning Bayesian network structures from continuous data. Master's thesis, Vanderbilt University, 2005.

I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.

I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46(1-3):389–422, 2002.

I. Guyon, C. Aliferis, and A. Elisseeff. Causal feature selection. In H. Liu and H. Motoda, editors, *Computational Methods of Feature Selection*. Chapman and Hall/CRC Press, 2007.

D. Hardin, I. Tsamardinos, and C. F. Aliferis. A theoretical characterization of linear SVM-based feature selection. In *Proceedings of the Twenty First International Conference on Machine Learning*, 2004.

D. M. Hausman and J. Woodward. Independence, invariance and the causal Markov condition. *British Journal for the Philosophy of Science*, 50:521–583, 1999.

G. H. John, R. Kohavi, and K. Pfleger. Irrelevant feature and the subset selection problem. In *Proceedings of the Eleventh International Conference on Machine Learning*, pages 121–129, 1994.

G. G. Judge, R. Carter Hill, W. E. Griffiths, H. Lütkepohl, and T.-C. Lee. *Introduction to the Theory and Practice of Econometrics, 2nd Edition*. Wiley, 1988.

R. Kohavi and G. H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97:273–324, 1997.

S. L. Lauritzen and D. J. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society, Series B*, 50:157–224, 1988.

P. Leray and O. François. BNT structure learning package, 2004. URL `banquiseasi.insa-rouen.fr/projects/bnt-slp/`.

D. Margaritis. Distribution-free learning of Bayesian network structure in continuous domains. In *Proc. of the 20th National Conf. on AI*, 2005.

D. Margaritis and S. Thrun. Bayesian network induction via local neighborhoods. In *Advances in Neural Information Processing Systems 12*, 1999.

C. Meek. Causal inference and causal explanation with background knowledge. In *Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence*, 1995.

R. Nilsson, J. M. Peña, J. Björkegren, and J. Tegnér. Consistent feature selection for pattern recognition in polynomial time. *The Journal of Machine Learning Research*, 8:589–612, 2007.

J. M. Peña, J. Björkegren, and J. Tegnér. Scalable, efficient and correct learning of Markov boundaries under the faithfulness assumption. In *Proceedings of the Eighth European Conference on Symbolic and Quantitative Approaches to Reasoning under Uncertainty*, pages 136–147, 2005.

J. Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, 2000.

J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, Los Altos, 1988.

J. Pearl. Causal diagrams for empirical research. *Biometrika*, 82(4):669–709, 1995.

J. Pearl and T. Verma. A theory of inferred causation. In *Proc. of the Second Int. Conf. on Principles of Knowledge Representation and Reasoning*. Morgan Kaufmann, 1991.

J.-P. Pellet and A. Elisseeff. A partial correlation-based algorithm for causal structure discovery with continuous variables. In *7th International Symposium on Intelligent Data Analysis*, 2007.

A. Raveh. On the use of the inverse of the correlation matrix in multivariate data analysis. *The American Statistician*, 39:39–42, 1985.

R. Scheines, P. Spirtes, C. Glymour, C. Meek, and T. Richardson. The TETRAD project: Constraint based aids to causal model specification. Technical report, Carnegie Mellon University, Dpt. of Philosophy, 1995.

A. J. Smola and B. Schölkopf. A tutorial on support vector regression. Technical report, NeuroCOLT2, 1998.

P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction, and Search, Second Edition*. The MIT Press, 2001. ISBN 0262194406.

A. Statnikov, D. Hardin, and C. F. Aliferis. Using SVM weight-based methods to identify causally relevant and non-causally relevant variables. Technical report, Vanderbilt University, USA, 2006.

D. Steel. Homogeneity, selection, and the faithfulness condition. Technical report, Michigan State University, Department of Philosophy, 2005.

M. Talih. *Markov Random Fields on Time-Varying Graphs, with an Application to Portfolio Selection*. PhD thesis, Hunter College, 2003.

R. Tibshirani. Regression shrinkage and selection via the lasso. Technical report, University of Toronto, 1994.

I. Tsamardinos and C. Aliferis. Towards principled feature selection: Relevancy. *Artificial Intelligence and Statistics*, 2003.

I. Tsamardinos, C. F. Aliferis, and A. Statnikov. Time and sample efficient discovery of Markov blankets and direct causal relations. In ACM Press, editor, *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 673–678, 2003.

I. Tsamardinos, L. E. Brown, and C. F. Aliferis. The max-min hill-climbing Bayesian network structure learning algorithm. *Machine Learning*, 2006.